

SISTEMA DE TECNIFICACIÓ DE JUDO AMB KINECT

Enginyeria Tècnica en Informàtica de Sistemes

Didac Rodríguez Sevillano



Data: 17 de juny de 2014
Director: Pau Fonseca Casas
Departament: Estadística i Investigació Operativa (EIO)
Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC)

ÍNDIX

CAPÍTOL 1 PRESENTACIÓ DEL PROJECTE	4
1.1 Introducció	4
1.2 Motivació	4
1.3 Objectius del projecte	5
CAPÍTOL 2 SISTEMES DE CAPTURA DE MOVIMENT	7
2.1 Sistemes òptics de captura de moviment	9
2.2 Sistemes no òptics de captura de moviment	10
CAPÍTOL 3 METODOLOGIA	13
3.1 El Hardware triat: La Kinect	13
3.1.1 Els orígens.....	13
3.1.2 Alternatives: Asus Xtion i PrimeSense Carmine.....	15
3.1.3 Hardware.....	15
3.1.4 Versions de Kinect i connexió amb l'ordinador	17
3.1.5 Funcionament de la Kinect.....	18
3.2 Eines de desenvolupament per Kinect.....	19
3.2.1 LibFreenect	19
3.2.2 OpenNI	19
3.2.3 SDK oficial de Microsoft.....	19
3.3 Driver i SDK de Microsoft	21
3.3.1 Skeleton, la clau de la detecció de persones	22
3.3.2 Sistema de reconeixement de veu	24
CAPÍTOL 4 ADAPTACIÓ DE KATACREATOR I KATATRAINGING.....	25

4.1 Proves d'entrenament amb Kinect	25
4.2 Alternatives per millorar el comportament de la Kinect	29
4.3 Conclusions de les proves.....	30
CAPÍTOL 5 VIRTUALDOJO	31
5.1 Millores de VirtualDojo respecte a KataCreator i KataTraining.....	31
5.2 Ús de VirtualDojo	33
5.2.1 Pantalla principal de VirtualDojo	34
5.2.2 Crear un model	35
5.2.3 Entrenar.....	36
5.2.4 Veure repetició entrenament i demo VirtualDojo	38
5.3 Classes més importants de VirtualDojo	39
5.3.1 Classe Model.....	39
5.3.2 Classe ItemModel	39
5.3.3 Classe Postura.....	40
5.3.4 Classe SkeletonInfo.....	41
5.3.5 Classe Dist.....	41
5.3.6 Classe CheckStance	41
CAPÍTOL 6 DISTRIBUCIÓ DE L'APLICACIÓ	43
6.1 Instal·lador ClickOnce.....	43
6.2 Distribució privada: Creació i configuració servidor propi	44
6.3 Distribució lliure: Codeplex	46
CAPÍTOL 7 PLANIFICACIÓ I PRESSUPOST.....	48
7.1 Planificació	48

7.2 Pressupost.....	49
7.2.1 Cost d'equipaments	49
7.2.2 Cost de personal	51
7.2.3 Cost total del projecte	51
7.3 Anàlisi DAFO	51
7.3.1 Debilitats	51
7.3.2 Fortaleses	52
7.3.3 Amenaces	52
7.3.4 Oportunitats	52
 CAPÍTOL 8 CONCLUSIONS.....	 53
8.1 Objectius coberts	53
8.2 Futur de VirtualDojo.....	53
 CAPÍTOL 9 REFERÈNCIES.....	 54
 CAPÍTOL 10 ÍNDEX DE FIGURES	 55
 CAPÍTOL 11 ÍNDEX DE TAULES.....	 57
 ANNEX: MANUAL DE VIRTUALDOJO	 58

Capítol 1 Presentació del projecte

1.1 Introducció

Aquest projecte neix com a continuació del PFC “Sistema de entrenamiento con Kinect”[1] realitzat pel Sr. Àngel Rodríguez Estévez sota la direcció del doctor Pau Fonseca i Casas, professor del departament d'Estadística i Investigació Operativa de la Universitat Politècnica de Catalunya.

El punt de partida del meu projecte són dues aplicacions relacionades però independents: una que permet crear un model de kates de karate (Kata Creator) i una altra per a poder entrenar-les posteriorment gràcies al sistema de detecció de persones Kinect de Microsoft (Kata Training).

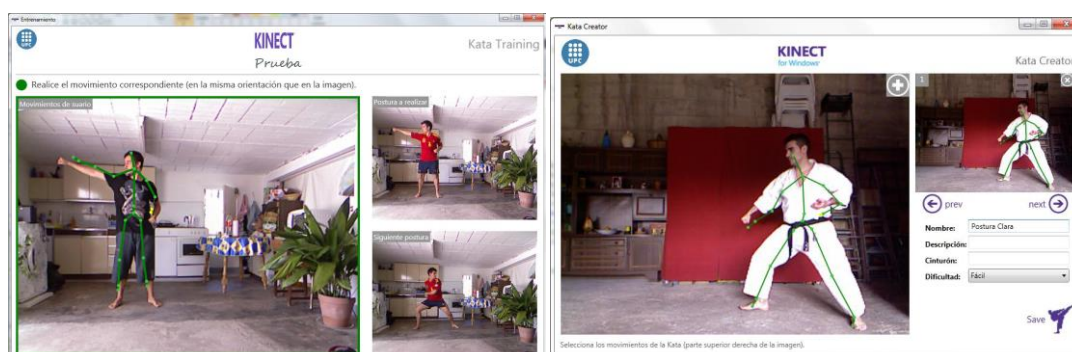


Figura 1.1 Kata Creator i Kata Training

Les kates de karate són seqüències establertes de cops i bloquejos llançats a l'aire que realitza una persona sola per tal de perfeccionar la seva tècnica.

1.2 Motivació

He practicat el judo des dels 6 anys d'edat, sóc cinturó negre 3er Dan i àrbitre nacional. Com a competidor he quedat un parell de cops campió de Catalunya Sènior i he participat en diversos campionats d'Espanya Universitaris representant a la UPC. El millor resultat el vaig aconseguir al 2005 quedant 2on al Campionat d'Espanya Universitari disputat a Granada representant a la FIB.

D'altra banda, des de fa un parell d'anys sóc el responsable d'informàtica de les competicions realitzades per la Federació Catalana de Judo i D.A., encarregant-me de tenir a punt els portàtils que es fan servir com a marcadors, dissenyar i configurar la xarxa i un servidor per tenir tota la competició automatitzada.



Figura 1.2 Trofeu Internacional Ciutat de Barcelona, exemple de competició informatitzada

Al conèixer l'oferta d'aquest PFC em va interessar molt perquè combinava la meua professió, la informàtica, amb la meua passió, el judo.

Em va semblar molt interessant poder crear un sistema d'entrenament pel meu esport i crec que, en cas d'obtenir uns resultats acceptables, tindria molt bona acollida dins del món del judo nacional que sempre està disposat a provar coses noves per tal de millorar els entrenaments.

1.3 Objectius del projecte

L'objectiu principal del projecte és intentar adaptar al món del judo les aplicacions realitzades pel Sr. Àngel Rodríguez. A priori, la gran diferència amb el karate és que a l'entrenament de judo es necessita sempre un company (Uke) i que les dues persones que realitzen l'activitat estiguin sempre en contacte.

En les primeres converses amb el tutor del projecte ja es posa de manifest que no serà fàcil aconseguir bons resultats amb la tecnologia de l'actual Kinect degut a que gairebé tots els entrenaments de judo es realitzen en parella i en contacte.

Inicialment vaig pensar en les kates de judo ja que els moviments són més lents que a d'altres tipus d'entrenaments que es poden realitzar en una sala de judo, però em vaig trobar amb el problema de que l'angle de visió que ens permet la Kinect és molt limitat. Al realitzar les kates de judo, la parella es desplaça per zones molt amples i marxaria del camp de visió que, com deia, és reduït.

Un cop descartades les kates pel camp de visió limitat de la Kinect, vaig decidir centrar-me en l'exercici estrella de qualsevol entrenament de judo: L'UchiKomi.

L'UchiKomi consisteix en repeticions d'una mateixa tècnica per tal de perfeccionar-la, acostumant el cos a fer-la instintivament, gairebé sense pensar. Es pot realitzar de manera estàtica o en moviment, provocant el desplaçament correcte del company i estudiant en cada repetició la posició del cos.

Un objectiu secundari del projecte és intentar donar una visió més comercial a l'aplicació.

Esquemàticament, els objectius del projecte serien:

- Aprenentatge i ús del llenguatge de programació c#. Tinc experiència programant en C i C++.
- Aprenentatge i ús de Windows Presentation Foundation (WPF). Tinc experiència en llenguatges similars com Visual Basic.
- Aprenentatge i ús del SDK oficial de Kinect i les eines que l'acompanyen com per exemple el KinectStudio.
- Entendre el funcionament de la Kinect per intentar adaptar KataCreator i KataTraining al món del judo.
- Entendre l'algoritme de reconeixement de postures i veure si es pot millorar.
- Fusionar KataCreator i KataTraining en una sola aplicació i donar-li un toc comercial.

Capítol 2 Sistemes de captura de moviment

La captura de moviments (també coneguda pels seus noms en anglès "motion capture" o simplement "mocap")[2] és una tècnica utilitzada per a emmagatzemar de manera digital moviments, generalment de persones o animals. Es basa en les tècniques de fotogrametria i és utilitzada principalment en el món del cinema, dels videojocs, dels esports o fins i tot amb finalitats mèdiques.

En el context de la producció d'una pel·lícula o d'un videojoc, es refereix a la tècnica d'emmagatzemar les accions d'actors i utilitzar aquesta informació per animar models digitals de personatges en animació 3D. Es pot utilitzar tant per capturar el moviment corporal total o parcial d'un actor com per capturar les seves expressions facials.



Figura 2.1 Captura de moviments facials utilitzada a la pel·lícula Avatar

En els esports d'alt nivell ja fa anys que s'utilitza la tecnologia en general per facilitar la feina del àrbitres a diversitat d'esports. Un dels casos més famosos es l'ull de falcó[3] del tennis que s'utilitza oficialment des de l'any 2006 per determinar si una bola dubtosa ha entrat als límits del terreny de joc o no. L'ull de falcó és un sistema de captura de moviment que, mitjançant les imatges captades per diverses càmeres ultra ràpides repartides pels voltants de la pista, reconeix la posició de la

pilota i determina si ha entrat o no gràcies al fet de tenir predefinides les dimensions de la pista.

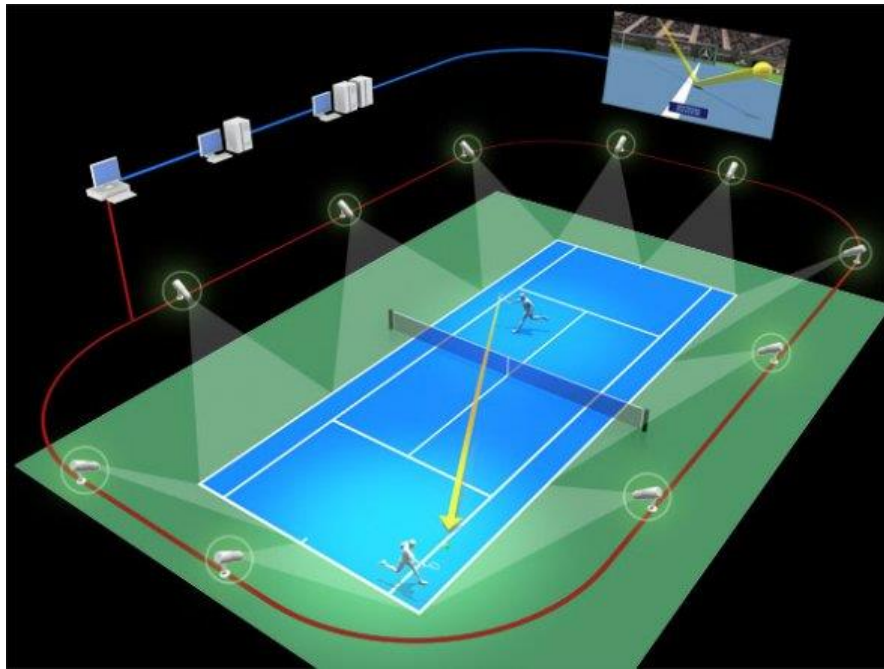


Figura 2.2 Ull de falcó, exemple de captura de moviment aplicat al tennis

En el món dels videojocs, a part de fer-se servir en l'animació 3D com al cinema, cada cop més s'intenten substituir els comandaments tradicionals com joysticks, pads amb botons, teclats i ratolins per sistemes de captura de moviments, com poden ser el Wimote de la Nintendo Wii, el comandament tauleta de la consola, també de Nintendo, WiiU, l'Eyetoy i el Move de la PlayStation de Sony o la Kinect de la Xbox de Microsoft. Tots aquest sistemes prioritzen tenir un cost reduït per a poder arribar a qualsevol usuari, enlloc d'una precisió tant fiable com els sistemes utilitzats al cinema o al tennis.



Figura 2.3 Diferents sistemes de captura de moviment enfocats als videojocs

Els sistemes de captura de moviment es poden dividir en dos grans grups: els sistemes òptics i els no-òptics.

2.1 Sistemes òptics de captura de moviment

Els sistemes òptics fan servir les dades capturades des d'un sensor d'imatges per triangular la posició de l'objecte o la persona entre dues o més càmeres calibrades per tal de poder realitzar un solapament de projeccions. Aquests sistemes fan servir marques per identificar els actors a capturar, les quals poden ser passives o actives. Les passives simplement són un punt reflectant a la roba de l'actor. Les actives emeten algun tipus de llum que pugui detectar el sensor. Els més utilitzats són sistemes de LEDS. Actualment els sistemes moderns ja són capaços de capturar persones sense l'ús d'aquestes marques.



Figura 2.4 Sistema de captura de moviment òptic amb marques actives

2.2 Sistemes no òptics de captura de moviment

En el camp dels sistemes de detecció de moviments també existeixen sistemes no òptics com poden ser els sistemes sobre inèrcia, els mecànics i els magnètics.

Els sistemes d'inèrcia fan servir giroscopis per tal de mesurar el moviment. Un avantatge d'aquests sistemes és el de no tenir punts morts ja que no depenen de visió directa, com en el cas de les càmeres, però d'altra banda són susceptibles a les interferències electromagnètiques ja que les dades capturades pels giroscopis s'acostumen a transmetre mitjançant tecnologies sense fils a l'ordinador que processa les dades. Un exemple de sistema per inèrcia seria el comandament Wiimote de la famosa consola de Nintendo Wii, que va ser la primera videoconsola que va apostar fort per la captura de moviments del jugador per a una millor experiència de joc.

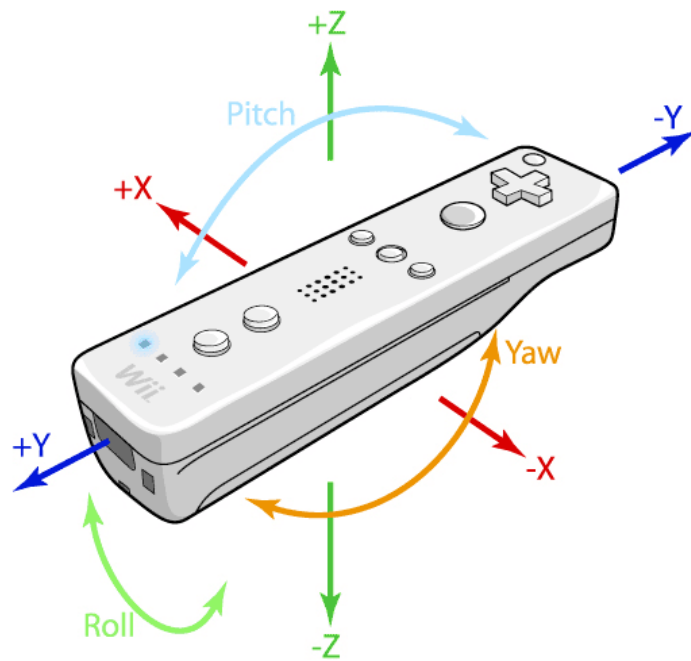


Figura 2.5 Wiimote, comandament de la videoconsola Wii de Nintendo.

Un altre tipus de sistemes de control de moviment són els mecànics. Els sistemes mecànics capturen directament el moviment del cos mitjançant sensors distribuïts per tot cos de l'actor com un exosquelet mecànic.



Figura 2.6 Sistema mecànic de captura de moviments

Encara que no tan utilitzats, també existeixen sistemes de captura de moviment magnètics que calculen la posició i l'orientació del cos mitjançant fluxos creats per bobines electromagnètiques.

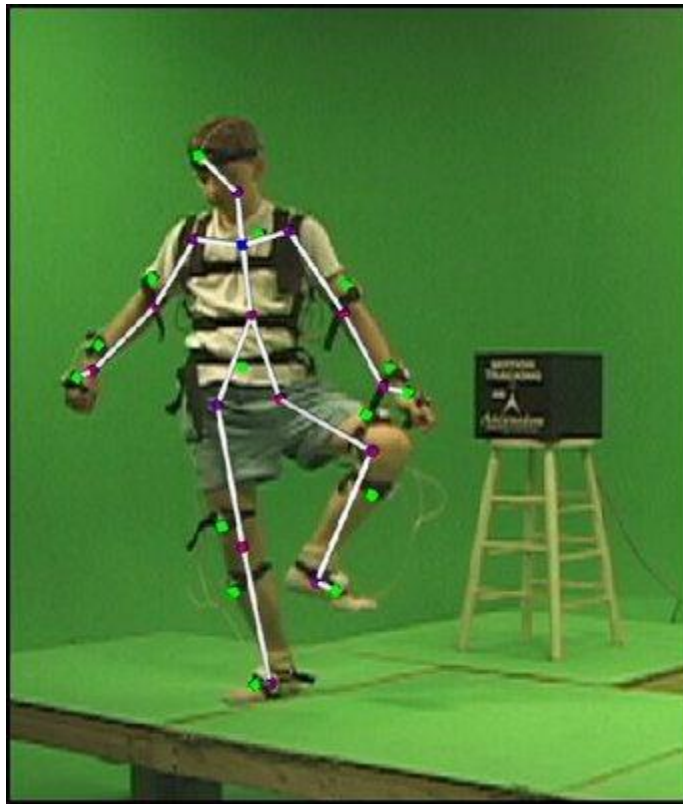


Figura 2.7 Sistema magnètic de captura de moviment

Capítol 3 Metodologia

3.1 El Hardware triat: La Kinect



Figura 3.1 Microsoft Kinect

3.1.1 Els orígens

El hardware més important en la realització d'aquest projecte és la Kinect. La Kinect és un dispositiu dissenyat l'any 2010 entre l'empresa israeliana PrimeSense i Microsoft.

PrimeSense[4] havia dissenyat un dispositiu, anomenat PrimeSensor, per controlar televisors directament amb les mans, sense l'ús de cap comandament a distància. Microsoft va voler portar aquest dispositiu al camp de la interacció amb videojocs en la seva videoconsola Microsoft Xbox 360.

Actualment Microsoft segueix apostant per aquest tipus de control i en la seva nova videoconsola, la Xbox One, ja incorpora de sèrie la segona revisió de hardware de la Kinect, coneguda comercialment com Kinect 2.0 o Kinect One. De moment Microsoft encara no ha publicat cap paquet de desenvolupament (SDK) per poder desenvolupar aplicacions Windows per a aquest nou dispositiu.



Figura 3.2 PrimeSensor

Les propietats més rellevants del PrimeSensor original van ser:

- Proveeix una interfície natural per a controlar dispositius electrònics.
- Els usuaris no necessiten disposar de cap complement o roba especial.
- La intensitat o coherència de les fonts de llum no són determinants per al seu funcionament.
- No requereix calibratge.

Microsoft va contractar a PrimeSense com a partner comercial per a que desenvolupés la Kinect i la va posar a disposició dels consumidors a un preu molt assequible i atractiu (el preu oficial de sortida van ser 150€). Aquest preu, tan baix si el comparem amb d'altres sistemes de reconeixement de moviment existents, va suposar que molts equips de desenvolupament arreu del món s'interessessin pel producte de cara a aprofitar funcionalitats diferents a la ideada inicialment per Microsoft, els videojocs. El preu ha estat determinant també per a l'elecció de la Kinect per aquest projecte final de carrera.

En vista d'aquest interès, la comunitat online va crear diversos drivers que mitjançant enginyeria inversa van aconseguir controlar el hardware de la Kinect i donar suport per a poder programar aplicacions per a la Kinect.

Microsoft va decidir aleshores crear una versió de la Kinect per a Windows amb un preu i característiques molt similars a la Kinect de la Xbox 360 i va publicar una primera versió del paquet de desenvolupament oficial (SDK) per a la Kinect a finals de febrer de 2012.

3.1.2 Alternatives: Asus Xtion i PrimeSense Carmine

L'èxit de la Kinect va portar a PrimeSense a buscar altres partners comercials per crear els seus propis dispositius de captura de moviments amb característiques semblants a la Kinect i intentar fer la competència a Microsoft en el camp del PC. Dos exemples són la Xtion d'Asus i la Carmine de la pròpia PrimeSense. Tots dos basats en el driver de codi obert OpenNi (veure 3.2.2).



Figura 3.3 Asus Xtion i PrimeSense Carmine, alternatives a Kinect

3.1.3 Hardware

La Kinect disposa d'una càmera d'imatge a color, dos sensors de profunditat, un vector de micròfons per captar el so i la seva procedència, un acceleròmetre per mesurar la inclinació de la càmera i un motor a la base que permet graduar la inclinació d'una manera automàtica dins d'un interval (de -27° fins a 27°).

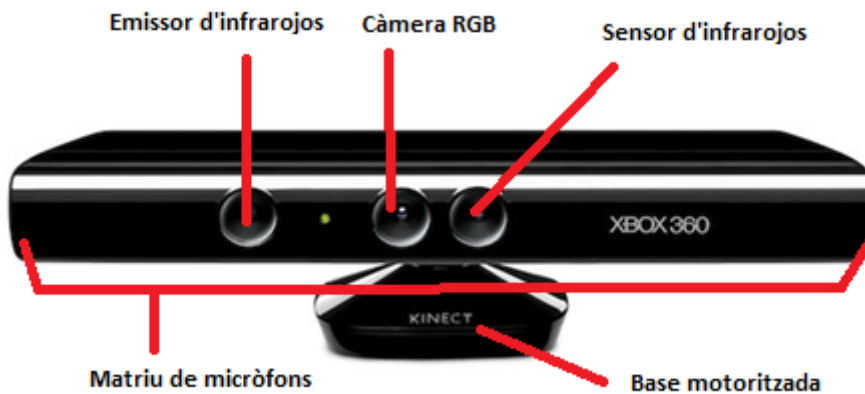


Figura 3.4 Components de la Kinect

Sensor de color	
Tipus de sensor	CMOS
Format	4:3
Resolució màxima	640x480
Frame rate (640x480)	30fps
Bits per píxel	32 bits

Sensor de profunditat	
Tipus de sensor	CMOS + projector d'IR
Format	5:4
Resolució màxima	1280x1024
Frame rate (640x480)	30fps
Bits per píxel	16
Rang	1.2m - 3.5m
Resolució espacial (2m de distància)	3mm
Resolució de profunditat (2m de distància)	1cm

Camp de visió	
Horitzontal	57°
Vertical	43°

Sistema d'àudio	
Nombre de micròfons	4
Sistema de cancel·lació de soroll	✓

Control d'inclinació	
Composició	motor + acceleròmetre (3 eixos)
Inclinació física (motor)	±27°

Processador	
Chip	PrimeSense PS1080-A2
Memòria	64MB DDR2 SDRAM

Taula 3.1 Especificacions mes rellevants de la Kinect

3.1.4 Versions de Kinect i connexió amb l'ordinador

La versió per Windows de Kinect té un connector USB 2.0 per a poder fer-la servir amb qualsevol ordinador actual però, en canvi, la versió de Xbox 360 de la Kinect necessita un adaptador per a transformar la connexió específica de la Kinect a un port USB 2.0 estàndard.



Figura 3.5 Connexió i adaptador de la Kinect

Les primeres versions de la Xbox 360 tampoc disposaven del port específic per a connectar la Kinect, cosa que va obligar a Microsoft a distribuir la Kinect amb l'adaptador per transformar la connexió a USB. Posteriorment Microsoft va realitzar una revisió de Hardware de la Xbox 360 anomenada Slim per ser més petita que la Xbox 360 original. La Xbox 360 Slim ja disposa de la connexió específica per a la Kinect i els packs de consola amb Kinect ja no incorporaven l'adaptador a USB 2.0

Cal remarcar que la versió de Kinect per a la Xbox 360 si es connecta amb l'adaptador USB2.0 a un ordinador es comporta exactament igual que la versió per a Windows, amb l'única excepció del mode "a prop" (Near Mode) que permet capturar moviment a una distància més reduïda (des de 40 cm).

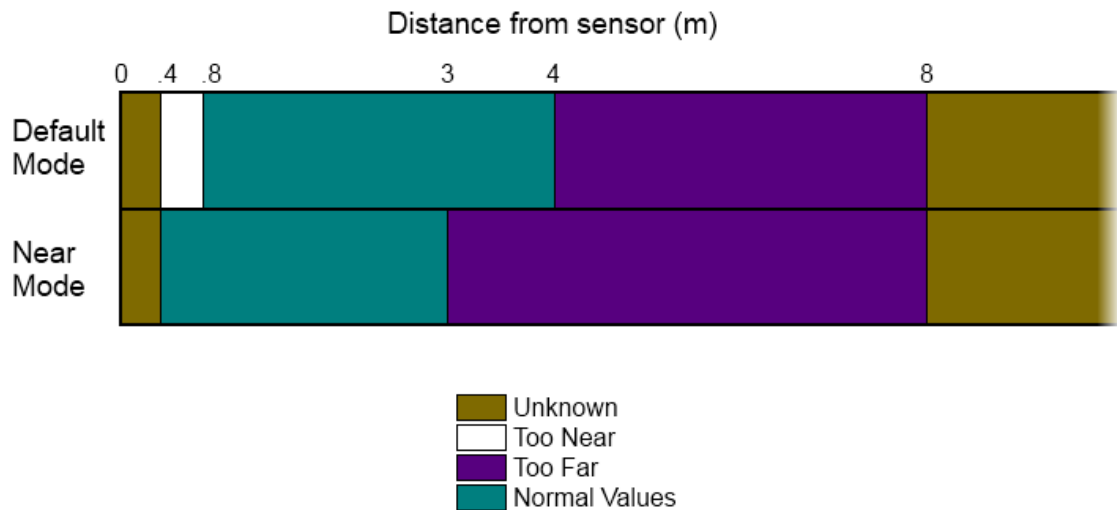


Figura 3.6 Diferencies entre el mode normal i el mode "a prop"

3.1.5 Funcionament de la Kinect

La clau del funcionament de la Kinect es troba en els seus sensors d'infrarojos. El funcionament dels mateixos és simple: per tal d'obtenir la profunditat dels objectes que es troben davant de la càmera Kinect, el projector d'infrarojos emet un patró irregular de rajos de diferents intensitats, repartint-los en el camp de visió de la càmera amb una distribució pseudo-aleatòria. La càmera d'infrarojos capta la distorsió del patró i, amb aquestes dades, reconstrueix el mapa de profunditat.



Figura 3.7 Rajos infrarojos emesos per la Kinect

3.2 Eines de desenvolupament per Kinect

A l'hora de desenvolupar aplicacions per la Kinect es disposen de diferents opcions de drivers i SDK per tal de controlar el dispositiu des d'un ordinador. A continuació exposaré les més conegudes.

3.2.1 LibFreenect

LibFreenect és un driver creat per una comunitat d'usuaris sense ànim de lucre anomenada OpenKinect[5]. Aquest driver va ser el primer en aparèixer i va sorgir mitjançant enginyeria inversa. Aquest driver permet obtenir les dades de baix nivell de Kinect, és a dir, la imatge a color i el mapa de profunditat únicament. LibFreenect també permet l'ús del motor d'inclinació de la Kinect per tal de graduar l'angle de visió.

3.2.2 OpenNI

OpenNI[6] és un driver creat per una organització sense ànim de lucre amb el mateix nom. Aquest driver deriva del codi proporcionat per PrimeSense pel seu dispositiu PrimeSensor i té el suport de la pròpia companyia. OpenNI, a part de proporcionar les dades de baix nivell de Kinect (imatge a color i mapa de profunditat), permet obtenir dades d'alt nivell, com són, els usuaris detectats i el seguiment dels seus esquelets. És compatible amb d'altres dispositius de captura de moviment com l'Asus Xtion i la Carmine de PrimeSensor. Amb la recent compra de PrimeSense per part d'Apple, la web oficial de OpenNi va ser tancada i actualment el projecte el porta l'empresa de desenvolupadors Structure.

3.2.3 SDK oficial de Microsoft

El Software Developer Kit (SDK)[7] oficial de Microsoft va aparèixer el 2012, dos anys més tard de que la Kinect sortís al mercat, degut al gran interès que va generar a la comunitat de desenvolupadors el potencial de la Kinect. Es compatible amb Windows 7 i Windows 8. El SDK inclou documentació, exemples i diverses eines per facilitar el desenvolupament amb la Kinect com el KinectStudio que permet enregistrar vídeos amb la Kinect per poder depurar el codi amb major comoditat.

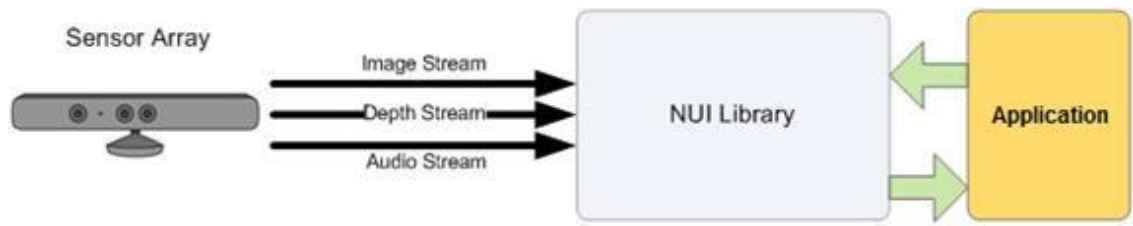


Figura 3.8 Interacció H/S de Kinect

Els llenguatges de programació suportats per les diferents versions de drivers són:

- SDK oficial de Microsoft: C++, C# i Visual Basic amb Microsoft Visual Studio.
- Alternatives OpenSource: Python, C, C ++, C #, Java, etc.

Pel que fa a la calibració de la Kinect, el driver de Microsoft calibra automàticament el dispositiu i ofereix uns resultats millors tant en precisió com en velocitat de detecció dels usuaris mentre que les alternatives OpenSource requereixen calibrar la Kinect seguint les instruccions proporcionades per cadascun dels drivers.

En general, el driver de Microsoft ofereix major precisió i més velocitat a l'hora de detectar els cossos dels usuaris. A més a més, ofereix totes les característiques que tenim a les alternatives OpenSource i d'una manera més documentada i senzilla. És per això que van ser triats per a la realització d'aquest projecte ja des dels seus orígens i no he trobat cap raó de pes per a fer servir cap de les alternatives OpenSource.

3.3 Driver i SDK de Microsoft

Com he explicat anteriorment, el SDK de Microsoft està pensat per desenvolupar aplicacions en C++, C# o Visual Basic. Com es lògic, el driver de Microsoft només permet treballar amb el seu SDK.

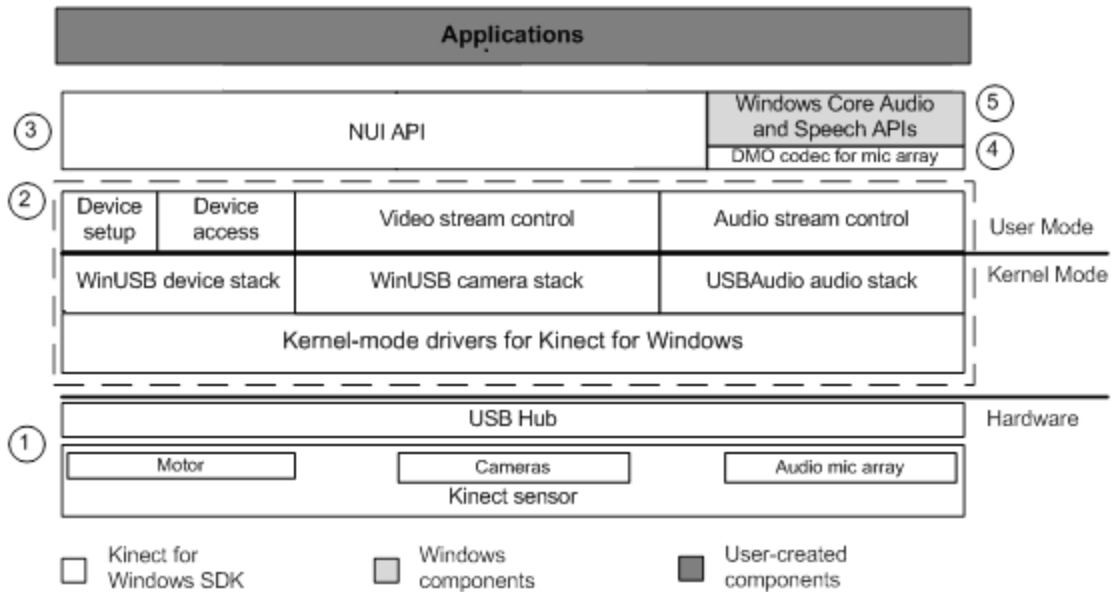


Figura 3.9 Arquitectura de la Kinect

1. Hardware - Components de la Kinect, inclosos els sensors, la matriu de micròfons, el motor d'inclinació i la connexió USB amb l'ordinador.

2. Controladors de la Kinect - S'instal·len amb el SDK de Microsoft i fan que Windows pugui detectar-la i utilitzar-la correctament.

3. Nui API - Conjunt d'API's que recopilen les dades dels sensors d'imatge i profunditat de la Kinect.

4. DirectX Media Object (DMO) - Control de veu a través de la matriu de micròfons.

5. Windows API estàndard - API de reconeixement de veu per a Windows.

A la taula 3.2 es poden veure algunes de les característiques més importants del driver de Microsoft des de la versió 1.5.

Sistema de seguiment d'esquelets	
Esquelets màxims detectats simultàniament	6
Articulacions per esquelet	20
Dades per articulació	Posició (en metres respecte al sistema de referència de la càmera) + orientació (absoluta o relativa)
Mode assegut	Detecció d'usuaris i seguiment de les 10 articulacions de braços i cap
Mode "a prop"	Permet la detecció d'objectes a 50 cm de la càmera (només disponible en la versió de Kinect per Windows)

Sistema d'àudio	
Reconeixedor de veu	Suporta Anglès, Francès, Castellà, Italià i Japonès

Altres característiques	
Reconeixedor de cares	Permet fixar una malla 3D a la cara dels usuaris i reconèixer gestos facials en temps real
Suport per a múltiples Kinect's	Permet l'ús de múltiples Kinect connectades al mateix ordinadors (en controladors USB diferents)

Taula 3.2 Característiques del SDK de Microsoft per a Kinect

3.3.1 Skeleton, la clau de la detecció de persones

L'API de la Kinect ens permet a cada frame detectar si hi ha algun esquelet (skeleton) a la imatge capturada. Aquesta funcionalitat és bàsica per al seguiment de persones i, en el cas d'aquest projecte, per a la comparació de postures entre el model i la imatge capturada a temps real mentre es realitza l'entrenament.

La Kinect és capaç de detectar fins a 6 persones a la vegada i de representar (track) l'esquelet de dues d'elles.

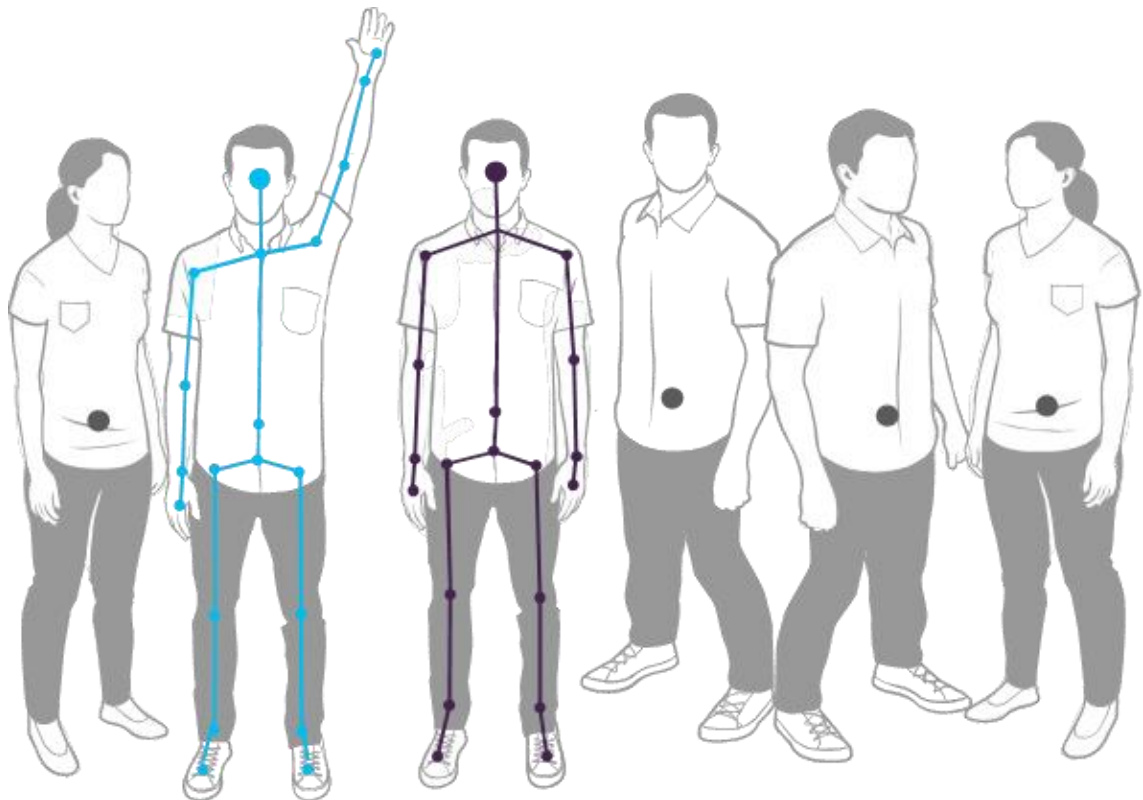


Figura 3.10 La Kinect pot reconèixer fins a 6 persones

Un esquelet, per a la Kinect, es un conjunt de 20 coordenades que representen a l'espai 20 parts del cos humà.

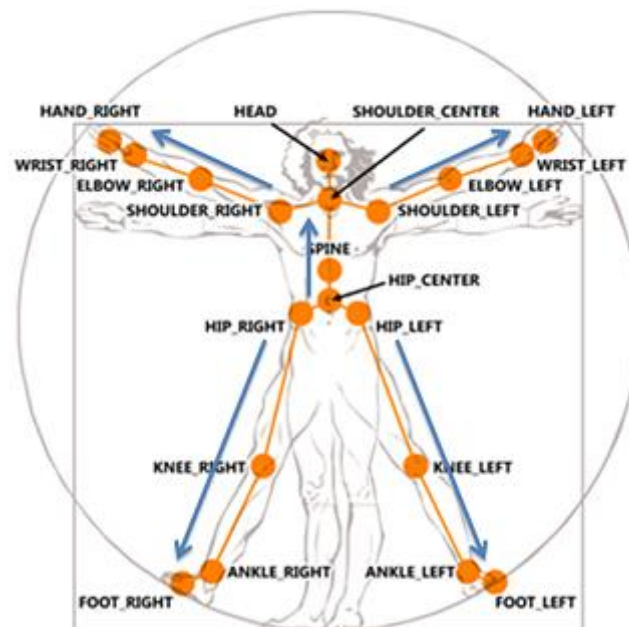


Figura 3.11 Els 20 punts que detecta l'esquelet de la Kinect

3.3.2 Sistema de reconeixement de veu

La Kinect disposa d'una matriu de micròfons per tal de poder detectar instruccions i saber d'on provenen espacialment. Per tal de detectar les ordres, Microsoft ha creat diferents paquets de reconeixement d'idiomes. Els idiomes disponibles són: anglès, francès, alemany, italià, japonès i castellà. Per aquesta raó i per arribar a un públic més ampli he decidit crear les aplicacions d'aquest projecte en castellà enlloc d'en català.

Capítol 4 Adaptació de KataCreator i KataTrainging

En aquest capítol explicaré les funcionalitats més importants de la Kinect i com han afectat a l'adaptació de KataCreator i KataTrainging al món del judo.

4.1 Proves d'entrenament amb Kinect

En una primera fase del projecte vaig intentar adaptar les aplicacions creades al PFC “Sistema de entrenamiento con Kinect” realitzat pel Sr. Àngel Rodríguez Estévez al judo.

Un cop estudiat el funcionament de la Kinect i analitzats diferents exemples tant de Microsoft com d'altres grups de desenvolupament independents, vaig fer alguns canvis al codi original per tal que el programa es comportés millor al tenir dues persones en contacte al davant.

El primer canvi va ser integrar un sistema, mitjançant codis d'exemple de Microsoft, per a que tingués només en compte l'esquelet més actiu de tots els que es capturessin. Aquest canvi suposava un augment del cost computacional bastant elevat i em va forçar a enregistrar els vídeos per processar-los a posteriori ja que el portàtil utilitzat no era prou potent.

Un altre canvi que vaig introduir per una major comoditat de l'ús de la Kinect va ser el control de l'angle d'inclinació per tal de graduar-la. Al fer proves de graduació de l'angle em vaig adonar d'una altra mancança important de l'aplicació original: la dependència d'una tercera persona que estigués pitjant el ratolí mentre s'utilitza el programa. Per solucionar aquest problema vaig decidir introduir el control de veu a la aplicació.

Un cop realitzats aquest canvis vaig realitzar diferents vídeos per tal de provar el sistema de reconeixement d'esquelets de la Kinect amb dues persones en contacte. Les proves es van enfocar als Uchikomis i a l'entrenament amb gomes. Els vídeos realitzats van ser:

- Uchikomi amb tori d'esquenes a la Kinect
- Uchikomi amb tori de cara a la Kinect
- Uchikomi amb tori i uke de costat fent la tècnica de cara a la Kinect
- Uchikomi amb tori i uke de costat fent moviment d'esquenes a la Kinect
- Uchikomi amb gomes amb Kinect darrera de tori
- Uchikomi amb gomes amb Kinect lateral de tori
- Uchikomi amb gomes amb Kinect davant de tori

Nota: Tori és la persona que realitza la Técnica i Uke és el company que l'ajuda a realitzar-la.

Per tots els vídeos es van fer servir judogis (vestit per practicar judo) de diferents colors (blau i blanc) entre Tori i Uke per a facilitar el reconeixement dels dos esquelets a la Kinect.

Els resultats obtinguts després d'aquestes primeres proves van ser:

- Si hi ha dues persones en contacte davant de la Kinect, el driver de Microsoft confon parts dels dos cossos constantment.
- El cost computacional de processar 30 imatges per segon i dels possibles esquelets detectats és molt alt. Si no es disposa d'un ordinador amb un processador prou potent es perden molts fotogrames i els resultats no són els esperats.
- En l'entrenament amb gomes es poden ajustar els valors de l'algoritme de comparació d'esquelets de KataTraing perquè es comptin les repeticions realitzades.



Figura 4.1 Uchikomi amb tori d'esquenes a la Kinect

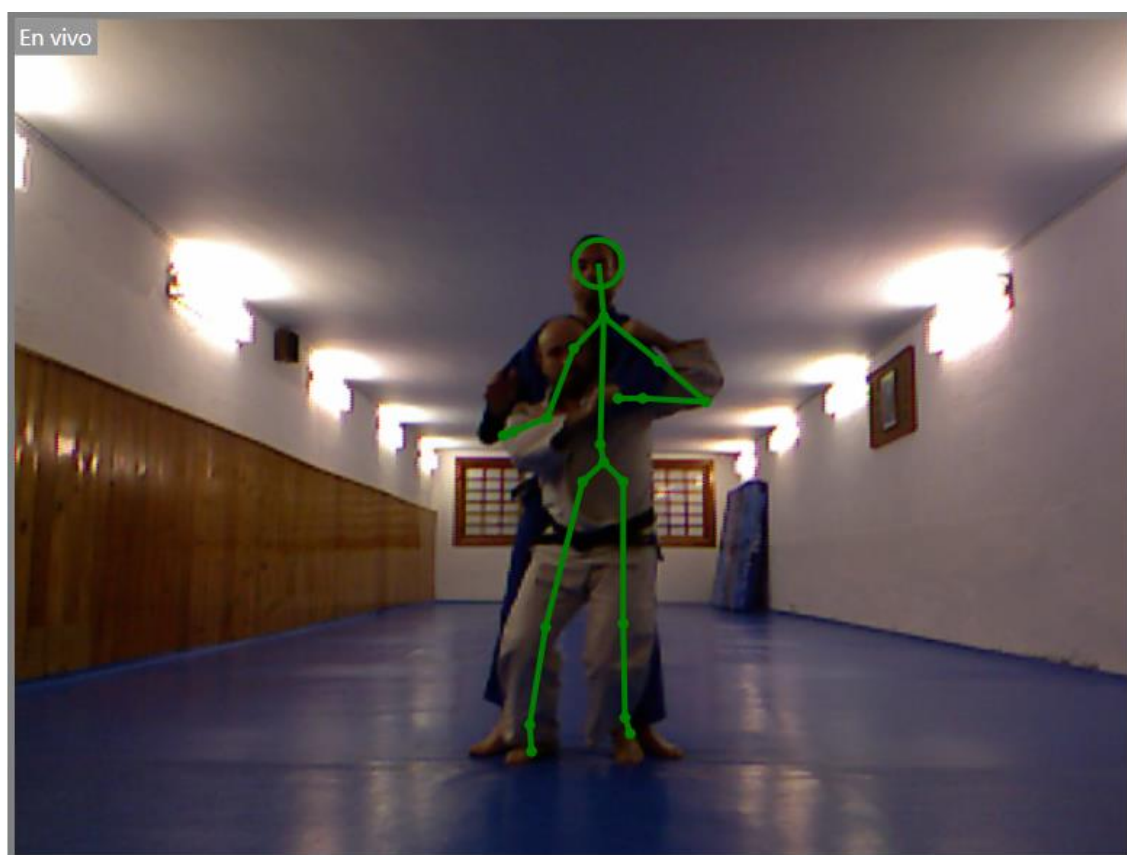


Figura 4.2 Uchikomi amb tori de cara a la Kinect



Figura 4.3 Uchikomi amb tori i uke de costat a la Kinect



Figura 4.4 Uchikomi de costat fent la tècnica d'esquenes a la Kinect

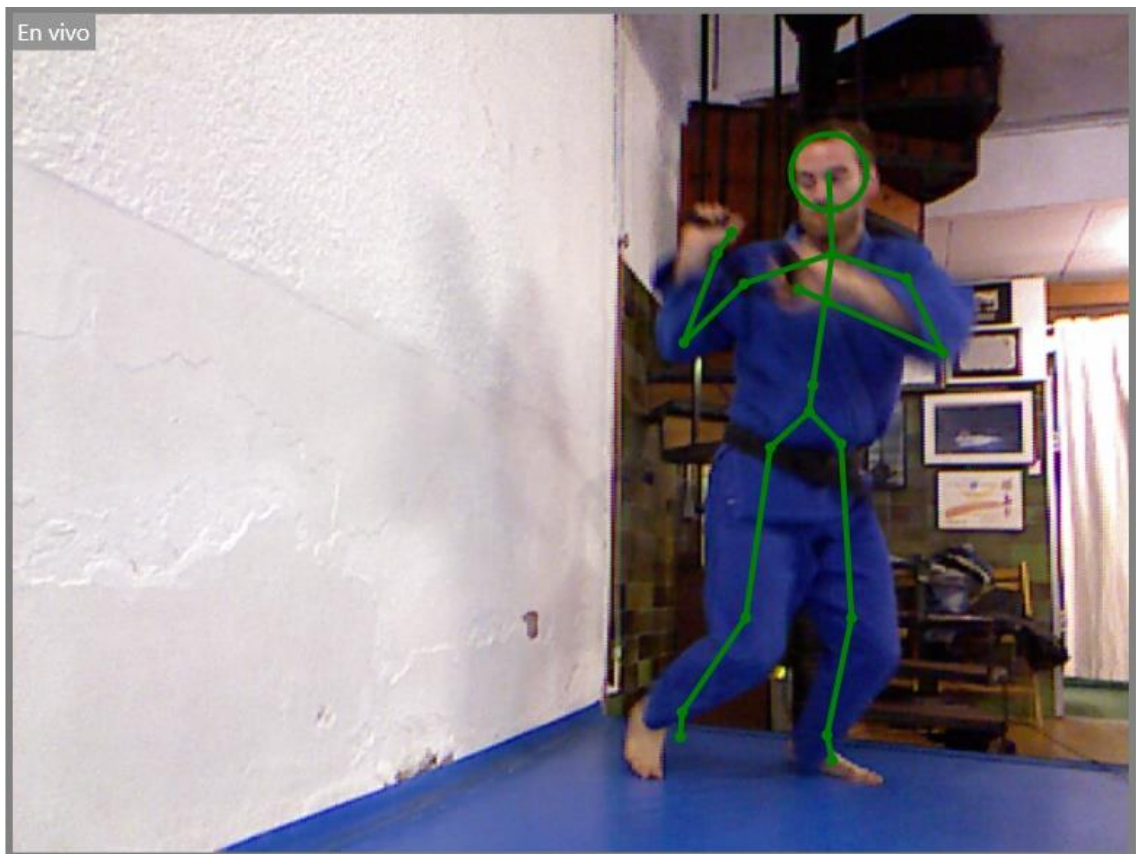


Figura 4.5 Uchikomi amb gomes amb Kinect darrera de tori

4.2 Alternatives per millorar el comportament de la Kinect

Al treballar amb el SDK oficial de Microsoft, els esquelets capturats per la Kinect venen donats sense que el programador pugui intervenir en el procés de detecció.

Un cop comprovat que el SDK de Microsoft no capturava correctament els esquelets al tenir dues persones en contacte va arribar el moment de buscar alternatives.

La primera alternativa era treballar amb dues Kinects. El SDK de Microsoft suporta treballar fins amb 4 Kinects simultàniament des de diferents angles. El fet de treballar amb múltiples Kinects ens permet eliminar punts morts però els resultats dels esquelets capturats quan les persones davant de les Kinects estan en contacte continuen sent deficientes ja que es capturen els esquelets per separat a cadascuna de les Kinects i a posteriori, mitjançant codi, es pot discriminar quina de les Kinects té menys punts morts. Dit d'una altra manera, el fet de tenir diverses Kinects no millora la qualitat dels esquelets capturats si estan en contacte, sino que es tenen les mateixes dades des de diferents punts de vista.

Un altre aspecte que em va fer descartar seguir el projecte enfocant-lo a l'ús de dues Kinects és que per tal que funcionin correctament han d'estar connectades a diferents hubs USB i la majoria de portàtils del mercat només disposen d'un.

Una altra alternativa era capturar el vídeo de l'entrenament per processar-lo a posteriori i evitar problemes d'ús de processador al treballar en temps real. Els resultats amb dues persones en contacte continuen éssent incorrectes i no ens permeten treballar correctament. A l'interessar-me per aquest punt vaig trobar un exemple de com enregistrar vídeo amb la Kinect i com reproduir-lo posteriorment. S'ha de tenir en compte que el SDK de Microsoft no incorpora cap tipus de procediment per enregistrar el vídeo, per tant hem de ser nosaltres com a programadors els que hem d'emmagatzemar al disc cadascuna de les imatges capturades per la Kinect i, en cas necessari, les dades dels esquelets i/o de la vista en profunditat.

4.3 Conclusions de les proves

Vista la poca llibertat que m'oferia el SDK de Microsoft a l'hora de modificar els mètodes que retornen els esquelets capturats per la Kinect i no trobar alternatives que permetessin millorar els resultats, el projecte s'enfocà a partir d'aquest moment a crear una sola aplicació general per a qualsevol art marcial on no hi hagi persones en contacte que permeti d'una manera còmoda crear un model amb una sèrie de postures i que pugui entrenar-lo a posteriori. Al parlar de comoditat, em refereixo a que sigui un entorn totalment controlable mitjançant ordres de veu i que l'aplicació respongui a aquestes ordres mitjançant sons conforme l'ordre ha estat processada correctament o no.

És en aquest punt on neix el programa VirtualDojo on deixaré de centrar-me en un esport en concret com el judo o el karate per fer una aplicació que pugui arribar a un públic més ampli al poder practicar qualsevol acció on hi hagi un sol usuari davant la Kinect i es pugui crear un model de postures.

Capítol 5 VirtualDojo

Un cop descartada la utilitat de la Kinect per a desenvolupar un sistema d'entrenament per judo, el projecte s'enfoca a fusionar i millorar les funcions de KataCreator i KataTraining en una sola aplicació que permeti realitzar un entrenament de qualsevol art marcial on no hi hagi contacte entre els usuaris.

Aquesta aplicació serà VirtualDojo on es podrà crear un model d'una sèrie de postures per a poder entrenar-les a posteriori definint el nombre de repeticions que es volen realitzar.

5.1 Millores de VirtualDojo respecte a KataCreator i KataTraining

Les millores implementades a VirtualDojo respecte a KataCreator i KataTraining resumides són:

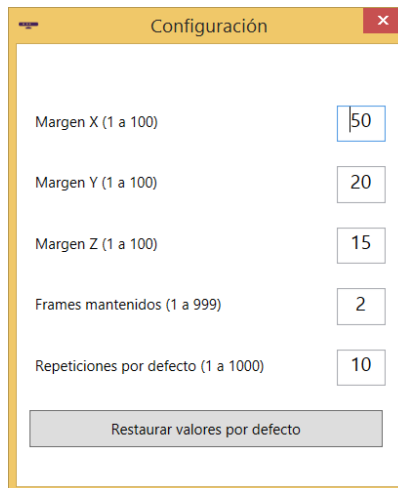
- Unificació en una sola aplicació amb control dels requisits previs.
- Control total de l'aplicació amb ordres de veu.
- Possibilitat d'enregistrar l'entrenament i reproduir-lo.
- Configuració dels marges per detectar si una postura és vàlida, el nombre de repeticions i els *frames* que volem que es mantingui una postura.
- Unificació dels models en un sol arxiu enlloc dels 3 que necessitava KataCreator.
- Actualització automàtica gràcies a la distribució des d'un servidor web.

VirtualDojo es pot controlar completament mitjançant ordres de veu per a una experiència més satisfactòria de l'usuari. Si es desitja també es poden desactivar les comandes de veu i utilitzar només el ratolí per evitar falsos positius en entorns on hi hagi soroll o s'estigui parlant entre les persones que estiguin realitzant l'entrenament.

VirtualDojo també permet enregistrar l'entrenament per a poder-lo veure un cop finalitzat. La implementació d'aquesta funcionalitat no és trivial, ja que el SDK de Microsoft no permet enregistrar vídeo de manera nativa i ho he aconseguit fent servir els exemples de la Kinect ToolBox[8] de com emmagatzemar cadascuna de les imatges a color capturades i les dades dels esquelets a un arxiu per poder llegir-lo i reproduir-lo a posteriori.

Una altra millora de VirtualDojo és la possibilitat de calibrar l'angle d'inclinació de la Kinect mitjançant la veu amb les ordres de veu "subir" i "bajar".

Durant les proves realitzades vaig poder comprovar que l'algoritme de comparació d'esquelets de KataTraining és prou fiable si el model s'ha creat correctament i, revisant el codi, vaig veure que funcionava dins d'uns marges preestablerts per cadascun dels eixos espacials X, Y i Z. VirtualDojo dóna l'opció a l'usuari d'ajustar els valors d'aquests marges donant-li l'opció de variar la precisió a l'hora de realitzar l'entrenament. A més d'aquest marges també es podran configurar el nombre de frames que volem que es mantingui una postura per donar-la com a vàlida i el nombre de repeticions per defecte a l'hora de realitzar un entrenament.



Configuración	
Margen X (1 a 100)	50
Margen Y (1 a 100)	20
Margen Z (1 a 100)	15
Frames mantenidos (1 a 999)	2
Repeticiones por defecto (1 a 1000)	10
Restaurar valores por defecto	

Figura 5.1 Valors configurables per l'usuari a VirtualDojo

5.2 Ús de VirtualDojo

En aquest apartat explicaré la lògica d'interacció de l'usuari amb l'aplicació VirtualDojo i cadascuna de les opcions que podrà trobar a l'utilitzar-lo en un entrenament.

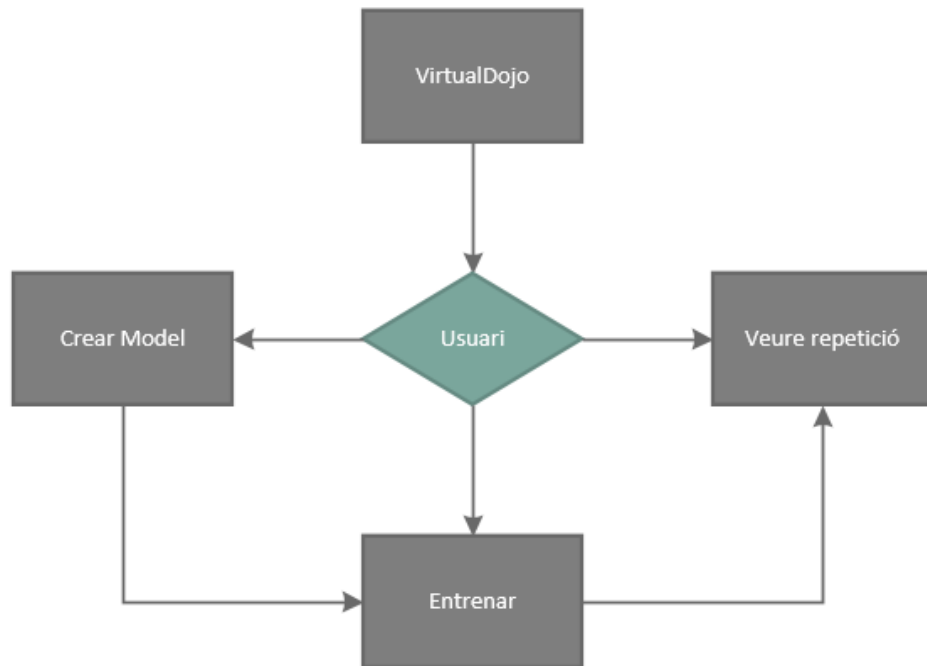


Figura 5.2 Interacció del usuari a VirtualDojo

5.2.1 Pantalla principal de VirtualDojo

Des de la pantalla principal del programa, l'usuari podrà triar amb el ratolí o amb una ordre de veu si vol crear un model, entrenar-lo o veure una repetició d'un entrenament ja realitzat. També tindrà l'opció de desactivar el reconeixement de veu a tota la aplicació i de configurar les opcions que he comentat anteriorment.

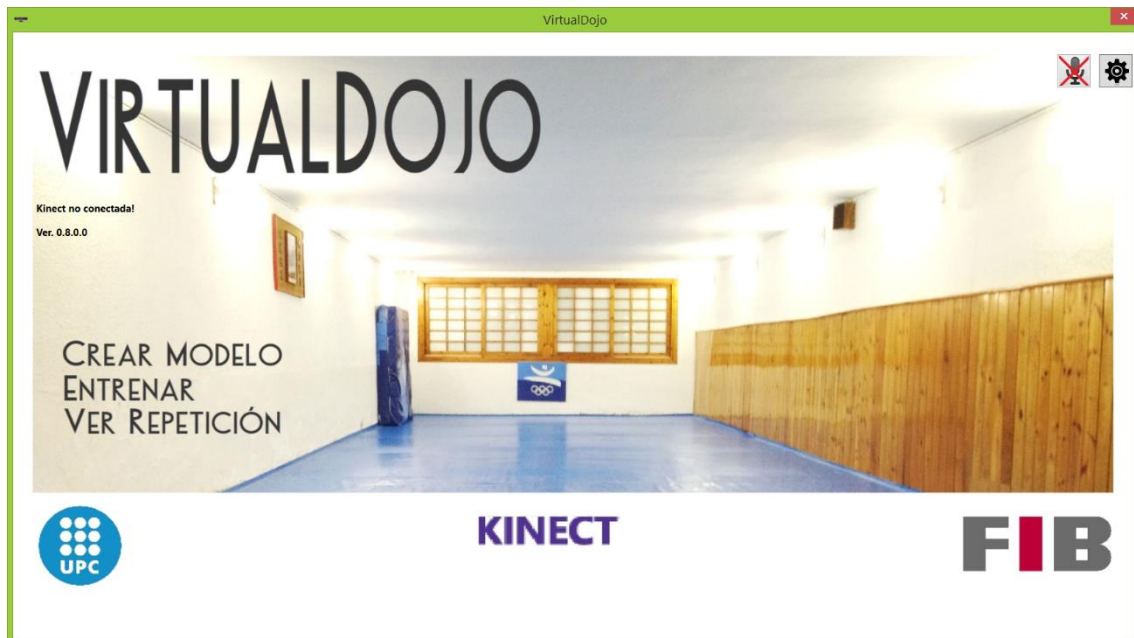


Figura 5.3 Pantalla principal de VirtualDojo

5.2.2 Crear un model

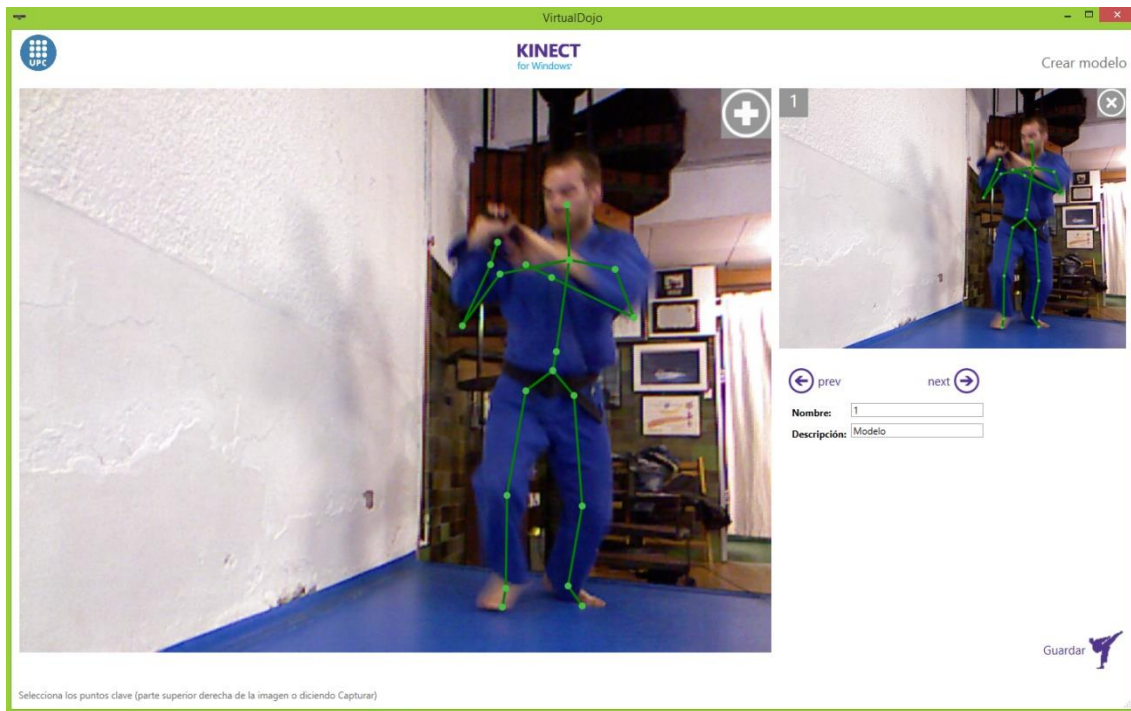


Figura 5.4 Pantalla de creació del model de VirtualDojo

Per a poder realitzar un entrenament primer s'ha de crear un model per a poder comparar-lo amb l'entrenament. En aquest apartat, per defecte es crearà el model amb nom **1** i amb **Modelo** com a descripció. Les comandes de veu reconegudes per l'aplicació en aquest apartat són:

- Subir, Bajar:** Per pujar i baixar respectivament l'angle de visió de la Kinect.
- Capturar:** Per afegir una postura al model. Si al fer la captura, l'esquelet detectat és correcte, s'emet un missatge de so informant a l'usuari. Si falten massa punts de l'esquelet (per defecte més de 3) o no se n'ha detectat cap, l'ordinador emet un so d'error.
- Guardar:** Si hi ha com a mínim una postura afegida al model, un nom i una descripció vàlides, es guarda al disc el model i s'emet un missatge de so conforme el model s'ha guardat correctament. En cas contrari s'informa a l'usuari que el model no ha pogut guardar-se correctament.

5.2.3 Entrenar

En aquest apartat l'usuari pot entrenar en temps real un dels models creats i pot enregistrar el seu entrenament per a poder estudiar-lo posteriorment.



Figura 5.5 Pantalla de selecció del model a Entrenar

A l'entrar a aquest apartat, el primer que veiem és una llista amb tots els models disponibles a la carpeta de treball del programa (VirtualDojo) dins del perfil personal de l'usuari.

L'usuari pot triar si vol entrenar o visualitzar el model. Les dues coses es poden seleccionar amb les comandes de veu "**Visualizar**" o "**Entrenar**". Si es dóna l'ordre per veu es prendrà per defecte el model de nom **1**.

També es poden triar mitjançant el teclat el nombre de repeticions que es volen realitzar del model durant l'entrenament. Per defecte el número de repeticions es de 10 i té un màxim de 1000 repeticions.



Figura 5.6 Pantalla d'entrenament de VirtualDojo

En la pantalla d'entrenament es pot donar la comanda de veu "**Empezar**" per a començar l'entrenament i enregistrar-lo al disc.

En la finestra de l'entrenament es mostren: la imatge en temps real de l'entrenament amb l'esquelet capturat, el nombre de repeticions realitzades, la postura actual a realitzar del model i la següent.

5.2.4 Veure repetició entrenament i demo VirtualDojo

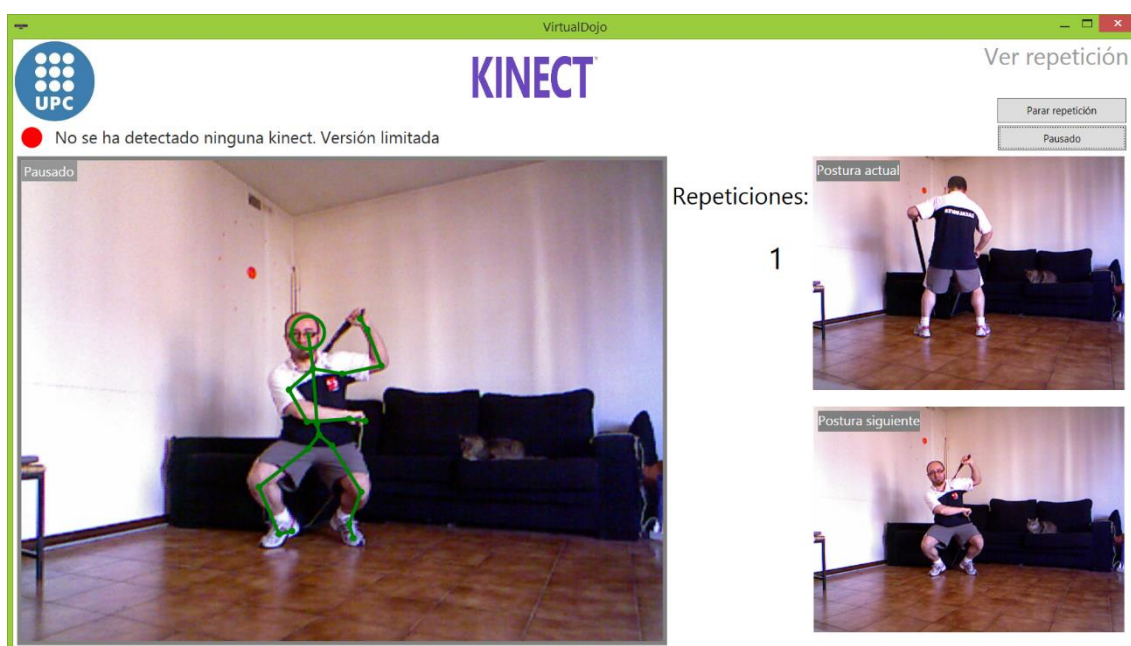


Figura 5.7 Pantalla de visualització de repeticions

En aquest apartat de VirtualDojo es pot veure una repetició d'un entrenament i comparar-la amb qualsevol model disponible. En primeres versions de VirtualDojo la funció de veure una repetició la vaig integrar a la finestra d'entrenament però en la part final del projecte va aparèixer la necessitat de poder accedir a una demostració de l'aplicació sense disposar d'una Kinect.

Per tal de mostrar l'esquelet a la pantalla es necessiten funcions del SDK de Microsoft que relacionen la imatge a color amb les dades de l'esquelet. Aquestes funcions depenen de l'objecte Kinect i en un principi feien que no fos viable la utilització de VirtualDojo sense tenir una Kinect connectada a l'ordinador. Per solucionar aquest problema en aquest cas puntual de reproduir un entrenament sense disposar d'una Kinect, vaig modificar el codi original per enregistrar les dades de l'objecte Kinect necessàries a la repetició en un arxiu al disc per fer-les servir a posteriori si no es disposa d'una Kinect. L'arxiu amb les dades del sensor, igual que els arxius amb imatges i els arxius de so, queden integrats a l'executable de VirtualDojo, ja que els vaig afegir al projecte de Visual Studio com a recursos de l'aplicació.

5.3 Classes més importants de VirtualDojo

En aquest apartat definiré les classes més importants de VirtualDojo i les relacions entre cadascuna d'elles. No he considerat oportú incloure en aquesta memòria classes menys importants, com les relacionades amb el tractament de finestres, de l'àudio, del vídeo o de la gestió d'arxius.

5.3.1 Classe Model

És la classe que representa un conjunt de postures ordenades que representen el model a reproduir. Conté mètodes per agregar una postura al Model, per eliminar-la i per moure's entre les postures, entre d'altres coses.

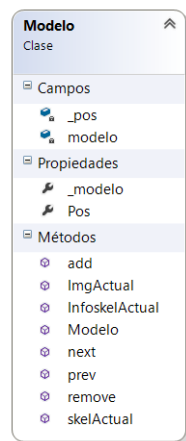


Figura 5.8 Classe Model

5.3.2 Classe ItemModel

Aquesta classe conté informació referent al model com ara el nom i la descripció. En futures versions de l'aplicació es poden afegir altres dades com dificultat, durada de l'entrenament, repeticions recomanades...

Aquesta informació es separa de la classe Model per facilitar la càrrega de la llista de models a diferents punts de l'aplicació sense carregar els models completament, amb totes les seves postures.

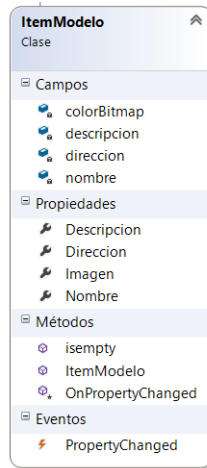


Figura 5.9 Clase ItemModelo

5.3.3 Clase Postura

La classe postura representa cadascun dels punts capturats al crear un model que després s'utilitzaran per a determinar si es realitza correctament l'entrenament, comparant-les a temps real amb cadascuna de les imatges capturades per la Kinect (30 imatges per segon).

Els atributs que conté són la imatge capturada, els esquelets que proporciona la Kinect i el càlcul d'informació necessària per a que l'algoritme de reconeixement de postures pugui comparar correctament les imatges.

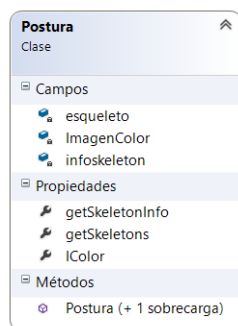


Figura 5.10 Clase Postura

5.3.4 Classe *SkeletonInfo*

Classe que conté les dades referents als esquelets que fa servir l'algoritme de reconeixement de postures. La informació que conté són distàncies de cada articulació del cos (joints) que ens proporciona la Kinect normalitzades respecte a la cintura.

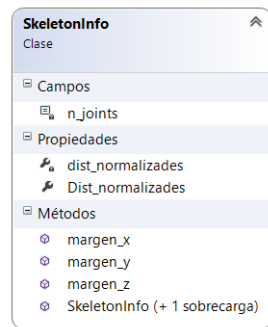


Figura 5.11 Classe *SkeletonInfo*

5.3.5 Classe *Dist*

Classe que conté la distància normalitzada d'una articulació del cos (Joint) respecte la cintura. A més a més de la distancia conté el nom del *joint* i si és un punt fiable o no (en anglès Inferred)

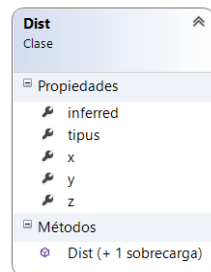


Figura 5.12 Classe *Dist*

5.3.6 Classe *CheckStance*

És potser la classe més important del projecte. Conté l'algoritme de comparació de postures que determinarà si l'esquelet capturat es considera igual a l'esquelet del model. Aquesta comparació es realitza cada cop que la Kinect detecta un esquelet correctament, podent arribar a un màxim de 30 vegades per segon, per tant el cost computacional de l'algoritme no pot ser molt elevat.

L'algoritme es basa en calcular la distància de cadascun dels diferents *joints* de l'esquelet que proporciona la Kinect normalitzada respecte a un mateix punt central per fer el seguiment dels usuaris. El punt triat per normalitzar les distàncies és la cintura.

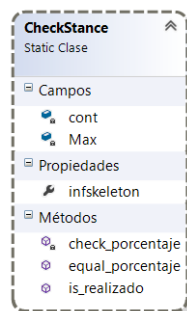


Figura 5.13 Classe CheckStance

Capítol 6 Distribució de l'aplicació

Un cop finalitzada una versió estable de VirtualDojo va arribar l'hora de decidir com distribuiria l'aplicació. En aquest capítol explicaré com vaig crear l'instal·lador de l'aplicació, la creació d'un servidor des d'on poder descarregar-lo, el suport a actualitzacions i una opció per crear un projecte OpenSource.

6.1 Instal·lador ClickOnce

Visual Studio permet publicar els instal·ladors del projecte mitjançant la tecnologia ClickOnce[9] creada per Microsoft. ClickOnce crea un instal·lador que conté les llibreries necessàries i que, si es configura correctament, detecta automàticament si hi ha actualitzacions disponibles al servidor web cada cop que s'inicia l'aplicació. Un altre avantatge de ClickOnce és que té cura de fer una instal·lació segura per l'usuari, aïllant l'aplicació perquè no tingui impacte en les llibreries del sistema operatiu i assegurant-se que no hi hagi problemes de permisos.

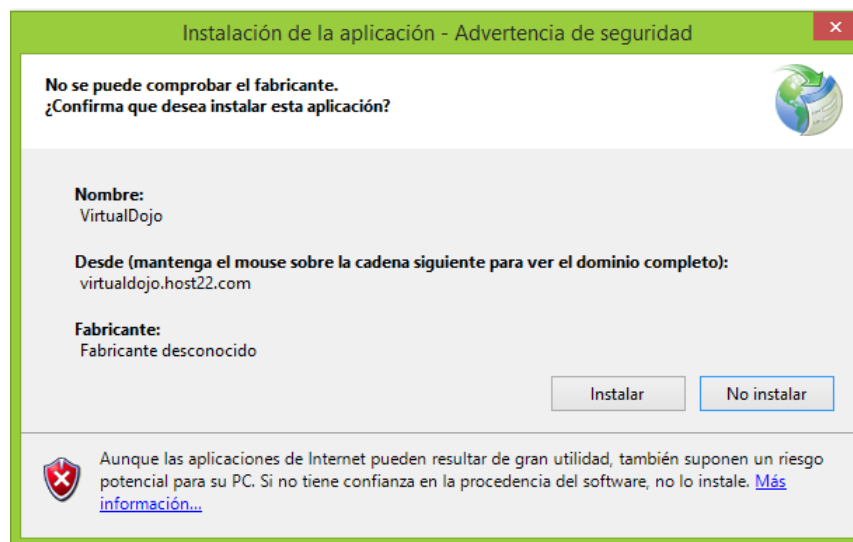


Figura 6.1 Instal·lador ClickOnce

L'instal·lador es pot crear localment, a una ruta de xarxa o directament sobre un servidor web. Com que a l'instal·lador creat amb ClickOnce s'hi havia d'afegir una demo, vaig crear primer un instal·lador per tal de crear després un executable que inclogués l'instal·lador, un model i una repetició demo, creant de passada la carpeta de treball de VirtualDojo.

6.2 Distribució privada: Creació i configuració servidor propi

Un cop creat l'instal·lador de l'aplicació, per donar un correcte servei de suport s'ha de crear un servidor web des d'on poder descarregar l'aplicació i les futures actualitzacions.

En cas de disposar d'un servidor web amb Windows, la configuració és automàtica gràcies a l'assistent de Visual Studio que es connecta directament amb el programa servidor web de Microsoft Internet Information Services (IIS)[10] amb totes les dependències, arxius i respectives associacions.

Al no disposar de cap servidor Windows, per crear el lloc web he triat el hosting gratuït de 000webhost.com que ofereix 1,5 GB d'espai a disc en un entorn Linux i una transferència mensual de 100 GB. El domini creat és virtualdojo.host22.com.

Per tal de poder executar l'instal·lador i les actualitzacions des d'un servidor web Linux cal definir els tipus MIME[11] del framework de .NET perquè es descarreguin correctament tots els paquets. Per definir-los vaig editar l'arxiu `.http_acces` del servidor com es pot veure a la figura 6.2.



```
File: /public_html/.htaccess
Normal textareas

# Do not remove this line, otherwise mod_rewrite rules will stop working
RewriteBase /

AddType application/x-ms-application application
AddType application/x-ms-manifest manifest
AddType application/octet-stream deploy
AddType application/octet-stream msu
AddType application/octet-stream msp
AddType application/octet-stream exe

AddType application/xaml+xml xaml
AddType application/x-ms-xbap xbap
```

Figura 6.2 Configuració tipus MIME del servidor web

Després de crear el domini s'ha de configurar Visual Studio per a publicar directament en ell, indicant la ruta del FTP del domini i la ruta des d'on s'instal·laran les actualitzacions.

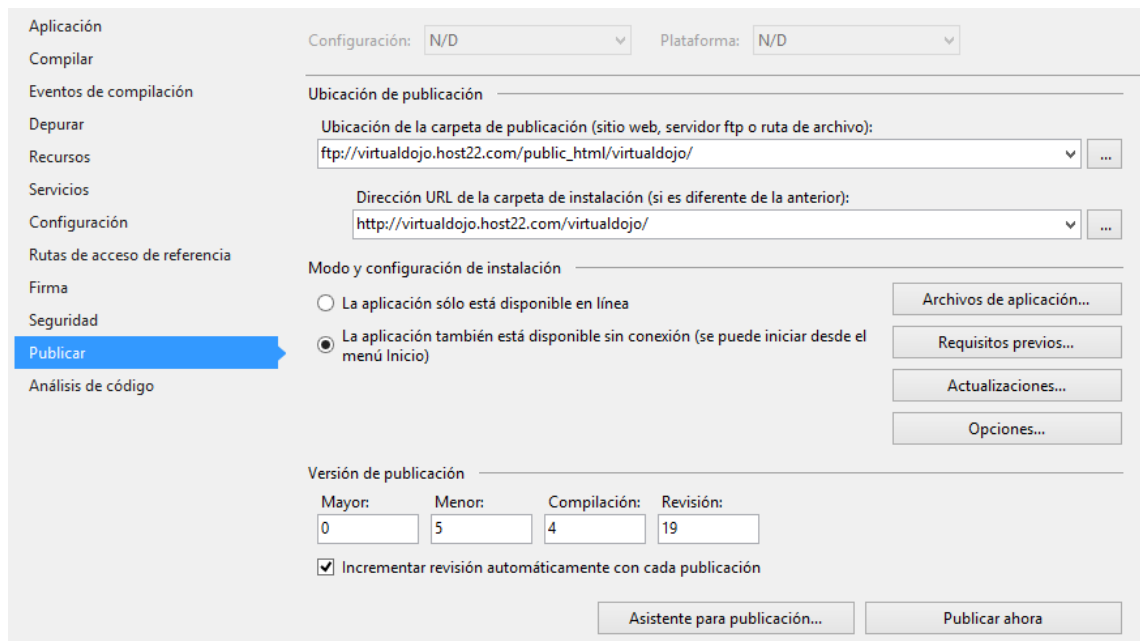


Figura 6.3 Configuració de Visual Studio per publicar al servidor web

Altres aspectes a configurar sobre l'instal·lador són:

- les llibreries necessàries per al reconeixement de veu i el funcionament de la pròpia Kinect.
- els requisits previs per al correcte funcionament de l'aplicació, en el cas de VirtualDojo són haver instal·lat la versió 4.5 de Microsoft .Net Framework, el SDK 1.8 de la Kinect i el paquet d'idioma castellà per a Kinect.
- que l'aplicació busqui actualitzacions a l'iniciar-se.
- que l'aplicació es pugui executar sense disposar de connexió a Internet.

Visual Studio 2013 ens crea una pàgina web per defecte amb enllaços a l'instal·lador de l'aplicació i al de la versió 4.5 de .NET. Vaig modificar la pàgina per incloure els enllaços als instal·ladors del SDK i del paquet d'idioma castellà mitjançant el editor HTML de codi obert Kompozer[12].

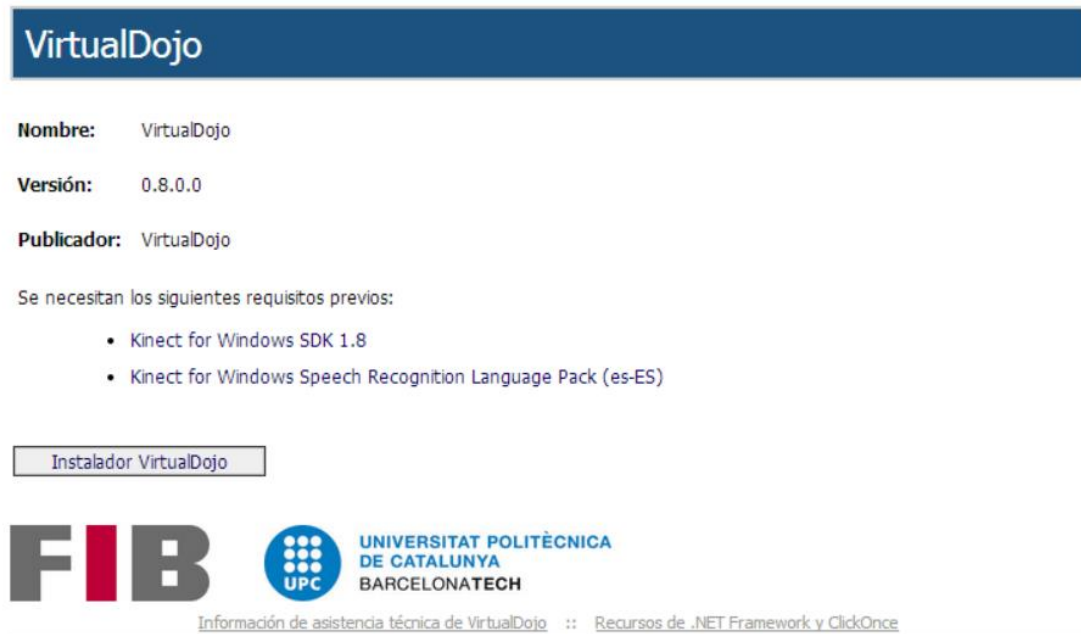



Figura 6.4 Web d'instal·lació i suport de VirtualDojo

6.3 Distribució lliure: Codeplex

En el cas d'alliberar el codi de l'aplicació i fer l'aplicació gratuïta, el mètode de distribució triat seria crear un projecte a la web Codeplex de Microsoft. Codeplex permet crear un projecte, pujar el codi directament des de Visual Studio, tenir un control de versions, crear un fòrum de discussió entre els desenvolupadors del projecte i tenir un repositori d'actualitzacions de l'aplicació per tal de que cada nova versió s'instal·li automàticament als clients que disposin de connexió d'internet al moment d'obrir VirtualDojo.

CodePlex
Project Hosting for Open Source Software
Register | Sign In |
Search all projects


Kinect Toolbox

HOME | SOURCE CODE | DOWNLOADS | DOCUMENTATION | DISCUSSIONS | ISSUES | PEOPLE | LICENSE

Page Info | Change History (all pages)
Follow (257) | Subscribe

Project Description

Kinect Toolbox is a set of useful tools for developing with Kinect for Windows SDK (1.7).

It includes helpers for gestures, postures, replay and drawing.

Blog post about Kinect Toolbox 1.2:
<http://blogs.msdn.com/b/eternalcoding/archive/2012/07/31/kinect-toolbox-1-2.aspx>

Blog post about Kinect Toolbox v1.1.1.0: <http://blogs.msdn.com/b/eternalcoding/archive/2012/02/01/official-kinect-for-windows-sdk-and-kinect-toolbox-1-1-1-are-out.aspx>

Blog post about Kinect Toolbox v1.1.0.2: <http://blogs.msdn.com/b/eternalcoding/archive/2011/11/05/kinect-for-windows-beta-2-is-out.aspx>

Blog post about Kinect Toolbox v1.1: <http://blogs.msdn.com/b/eternalcoding/archive/2011/08/02/kinect-toolkit-1-1-template-based-posture-detector-and-voice-commander.aspx>

Blog post about Kinect Toolbox gestures detection :
<http://blogs.msdn.com/b/eternalcoding/archive/2011/07/04/gestures-and-tools-for-kinect.aspx>

NuGet package: <http://nuget.org/List/Packages/KinectToolbox>

List of features:

Gestures

- » *SwipeGestureDetector* can detect the following gestures
 1. *SwipeToLeft*
 2. *SwipeToRight*
- » *TemplatedGestureDetector* can record gestures in a learning machine and detect them later. Using this class, you can detect every kind of 2D gestures (letters, geometric shapes such as circle...)

Postures

- » *AlgorithmicPostureDetector* can detect the following postures:
 1. None
 2. HandsJoined
 3. LeftHandOverHead
 4. RightHandOverHead
 5. LeftHello
 6. RightHello

Search Wiki & Documentation


download

CURRENT Kinect Toolbox 1.3


DATE Tue Apr 30, 2013 at 7:00 AM


STATUS Stable

DOWNLOADS 4.322

RATING  2 ratings
[Review this release](#)

MOST HELPFUL REVIEWS

 put a newer article on latest version up explaining *.save files, how to load them, and how to save them. If it wasn't for that you ... [\(more\)](#)

 It's really a useful Tools for developers.

[View all reviews](#)

ACTIVITY

PAGE VIEWS	VISITS	DOWNLOADS
674	283	109

Days: 7 30 All [Details](#)

Figura 6.5 Exemple de proiecte creat a Codeplex

Capítol 7 Planificació i pressupost

7.1 Planificació

Les converses per iniciar el projecte es van iniciar al maig de 2013, mes en el que vaig realitzar la inscripció del mateix. El fet de no matricular-lo al juliol de 2013 va determinar tenir més temps disponible per a la realització del projecte amb vistes de presentar-lo al segon quadrimestre del curs 2013-2014.

La planificació per mesos ha estat la següent:

Juliol - Setembre 2013: Estudi de la viabilitat del projecte, lectura PFC previ, entendre el codi de Kata Creator i Kata Training, aprenentatge de c# sobre WPF i estudi d'exemples i possibilitats de la Kinect.

Octubre - Novembre 2013: Modificació del codi de Kata Creator i Kata Training perquè només reconegui l'esquelet més actiu, es pugui controlar amb la veu, permeti graduar la inclinació de la càmera i enregistri l'entrenament si es desitja.

Desembre 2014: Proves d'entrenament de judo i buscar alternatives per la correcta detecció de dos esquelets.

Gener - Febrer 2014: Creació de VirtualDojo unificant les dues aplicacions en una sola implementant les millores detectades.

Març - Abril 2014: Crear entorn de distribució de VirtualDojo i realització de proves

Maig - Juny 2014: Documentació del projecte.

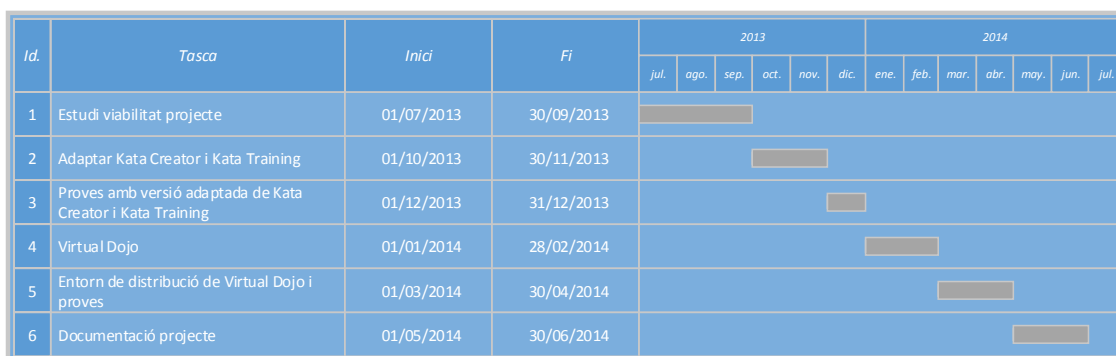


Figura 7.1 Diagrama de Gantt del projecte

7.2 Pressupost

Aquesta secció presenta una estimació econòmica del cost que hauria tingut aquest projecte si no hagués estat desenvolupat en l'àmbit d'un projecte final de carrera. Aquesta anàlisi té en compte tant el cost dels equipaments necessaris com el cost de personal.

7.2.1 Cost d'equipaments

El costos dels equipaments necessaris per a la realització d'aquest projecte els podem dividir en costos de Hardware, de Software i del lloguer de la oficina.

Pel que fa al Hardware, es necessita un ordinador portàtil de gama mitjana i una Kinect tant per al desenvolupament del software com per a la realització de proves i demos a possibles clients.

Hardware	Quantitat	Preu per unitat	Cost total
Ordinador gama mitjana	1	700€	700€
Kinect	1	100€	100€
TOTAL			800€

Taula 7.1 Costos de Hardware

Pel que respecta al software he tingut en compte el preu d'ampliació a una llicència Professional de la llicència de Windows 8 Home que ve inclosa al preu de l'ordinador de gama mitjana. Un altre cost de software seria la llicència de Visual Studio 2013 que és el paquet de desenvolupament utilitzat en el projecte.

Software	Quantitat	Preu per unitat	Cost total
Visual Studio Professional 2013	1	650€	650€
Windows 8 Professional	1	150€	150€
TOTAL			800€

Taula 7.2 Costos de Software

Un altre cost que he tingut en compte és el lloguer d'una oficina petita on realitzar el projecte. Tenint en compte que el temps total de dedicació al projecte es d'unes 350 hores i una jornada laboral de 40 hores setmanals, si el projecte s'hagués realitzat a jornada completa s'haurien necessitat 3 mesos per finalitzar-lo.

Concepte	Durada	Cost mensual	Cost total
Lloguer oficina	3 mesos	600€ mensuals	1.800€
Llum	3 mesos	50€ mensuals	150€
Aigua	3 mesos	20€ mensuals	60€
Internet	3 mesos	30€ mensuals	90€
TOTAL			2.100€

Taula 7.3 Costos lloguer oficina

Aleshores, el total dels costos d'equipament al sumar el costos de Hardware, Software i lloguer de l'oficina és de 3.700€ .

7.2.2 Cost de personal

Per calcular el cost de personal he tingut en compte el temps dedicat i tres rols separats, el d'analista a un preu de 50 €/hora, el de programador a un preu de 30 €/hora i el de tècnic de sistemes també a 50€/hora.

Tasca	Hores	Rol	Cost
Estudi viabilitat del projecte	15	Analista	750€
Adaptar KataCreator i KataTraining	85	Programador	2.550€
Proves amb versió adaptada aplicacions	10	Programador	300€
Creació de VirtualDojo i proves	80	Programador	2.400€
Entorn de distribució	15	Tècnic de sistemes	750€
Documentació del projecte	150	Programador	4.500€
TOTAL	355		11.250€

Taula 7.4 Costos de personal

7.2.3 Cost total del projecte

El cost total del projecte, sumant el cost total dels equipaments i el cost de personal es de 14.950€

7.3 Anàlisi DAFO

En el punt actual del projecte, havent creat una aplicació funcional, he considerat oportú realitzar un anàlisi de la situació actual de la aplicació i de la viabilitat de poder-la comercialitzar. Queda fora del abast del projecte desenvolupar un pla de negoci complert, però sí que he cregut convenient incloure un anàlisi DAFO. A continuació exposaré les diferents debilitats, amenaces, fortaleces i oportunitats de VirtualDojo.

7.3.1 Debilitats

Les debilitats que presenta VirtualDojo són:

- Poca precisió dels sensors de la Kinect. El reconeixement d'esquelets té problemes si la il·luminació de la sala no es la correcta.
- El format de les repeticions no segueix cap estàndard de video per tant només es poden reproduir amb VirtualDojo.
- Les imatges capturades tenen una resolució molt baixa (640x480) si les comparem amb les resolucions més altes utilitzades avui dia.

7.3.2 Fortaleses

Les fortaleses de VirtualDojo són:

- El baix cost del sistema, una Kinect i un ordinador actual de gama mitjana, el pot fer molt atractiu a l'usuari.
- L'ús intuïtiu de l'aplicació la fa arribar a un públic més ampli.
- No existeix al mercat actualment cap aplicació semblant, que permeti crear a l'usuari els seus propis models, que pugui ser competència.

7.3.3 Amenaces

Les amenaces de VirtualDojo poden ser:

- El creixement de les alternatives de hardware semblant a la Kinect que són OpenSource. Amb l'actual *boom* de les impresores 3D, els sistemes com la Kinect o el Xtion d'Asus s'estan extenent com a dispositius *low cost* per a fer escanejos 3D.
- Facilitat de còpia de l'aplicació per part de programadors experimentats en l'ús de Kinect i de sistemes semblants de captura de moviments.
- Aplicacions existents d'entrenaments esportius que es limiten a fer repetir a l'usuari models pre-definits.

7.3.4 Oportunitats

Les oportunitats futures de VirtualDojo són:

- Suport i actualitzacions de l'aplicació de pagament.
- Ofereix versions a mida de l'aplicació i/o amb altres idiomes disponibles.
- Escalabilitat del projecte migrant el codi a la nova Kinect One de Microsoft.

Capítol 8 Conclusions

Per finalitzar aquesta memòria, en aquest capítol explicaré els objectius coberts durant el projecte, els aspectes bons i dolents que m'he trobat al treballar amb la Kinect, sobre el futur de VirtualDojo i les expectatives personals complertes durant la realització d'aquest projecte final de carrera.

8.1 Objectius coberts

En la realització d'aquest projecte final de carrera he trobat molt enriquidor treballar amb l'última versió del paquet de Microsoft Visual Studio, la creació senzilla d'interfícies d'usuari amb Windows Presentation Foundation, conèixer el funcionament de la Kinect i del seu SDK, aprendre un llenguatge de programació com C# i, ja més relacionat amb l'administració de sistemes informàtics, la creació d'un espai web des d'on fer la distribució de l'aplicació.

En canvi, va ser una mica frustrant que l'única aplicació funcional relacionada amb el meu esport, el judo, fos l'entrenament amb gomes, ja que no és un entrenament tant tècnic com l'Uchikomi si no que és un entrenament físic, enfocat a muscular i suar més que a perfeccionar una tècnica de projecció.

8.2 Futur de VirtualDojo

Crec que el futur de VirtualDojo vindrà donat per l'interés dels primers usuaris del programa i les demandes de canvis que puguin aportar. També serà interessant veure les millores en la detecció d'esquelets que aportarà la nova Kinect One i el SDK 2.0 de Microsoft que està previst que aparegui durant l'estiu d'aquest any 2014.



Figura 8.1 Kinect One

Capítol 9 Referències

- [1] <http://upcommons.upc.edu/pfc/bitstream/2099.1/18626/1/85859.pdf>
- [2] http://en.wikipedia.org/wiki/Motion_capture
- [3] <http://www.hawkeyeinnovations.co.uk/>
- [4] <http://www.primesense.com/>
- [5] <http://openkinect.org/>
- [6] <http://structure.io/openni>
- [7] <http://www.microsoft.com/en-us/kinectforwindows/>
- [8] <http://kinecttoolbox.codeplex.com>
- [9] <http://en.wikipedia.org/wiki/ClickOnce>
- [10] [http://es.wikipedia.org/wiki/Internet Information Services](http://es.wikipedia.org/wiki/Internet_Information_Services)
- [11] <http://robindotnet.wordpress.com/2010/06/12/mime-types-for-clickonce-deployment/>
- [12] <http://kompozer.net/>

Capítol 10 *Índex de figures*

Figura 1.1 Kata Creator i Kata Training	4
Figura 2.1 Captura de moviments facials utilitzada a la pel·lícula Avatar	7
Figura 2.2 Ull de falcó, exemple de captura de moviment aplicat al tennis	8
Figura 2.3 Diferents sistemes de captura de moviment enfocats als videojocs ...	9
Figura 2.4 Sistema de captura de moviment òptic amb marques actives	10
Figura 2.5 Wiimote, comandament de la videoconsola Wii de Nintendo.	11
Figura 2.6 Sistema mecànic de captura de moviments	11
Figura 2.7 Sistema magnètic de captura de moviment	12
Figura 3.1 Microsoft Kinect.....	13
Figura 3.2 PrimeSensor	14
Figura 3.3 Asus Xtion i PrimeSense Carmine, alternatives a Kinect	15
Figura 3.4 Components de la Kinect.....	15
Figura 3.5 Connexió i adaptador de la Kinect	17
Figura 3.6 Diferències entre el mode normal i el mode "a prop"	18
Figura 3.7 Rajos infrarojos emesos per la Kinect.....	18
Figura 3.8 Interacció H/S de Kinect.....	20
Figura 3.9 Arquitectura de la Kinect	21
Figura 3.10 La Kinect pot reconèixer fins a 6 persones.....	23
Figura 3.11 Els 20 punts que detecta l'esquelet de la Kinect.....	23
Figura 4.1 Uchikomi amb tori d'esquenes a la Kinect	27
Figura 4.2 Uchikomi amb tori de cara a la Kinect	27
Figura 4.3 Uchikomi amb tori i uke de costat a la Kinect.....	28
Figura 4.4 Uchikomi de costat fent la tècnica d'esquenes a la Kinect	28
Figura 4.5 Uchikomi amb gomes amb Kinect darrera de tori	29
Figura 5.1 Valors configurables per l'usuari a VirtualDojo	32
Figura 5.2 Interacció del usuari a VirtualDojo	33
Figura 5.3 Pantalla principal de VirtualDojo.....	34
Figura 5.4 Pantalla de creació del model de VirtualDojo	35
Figura 5.5 Pantalla de selecció del model a Entrenar.....	36
Figura 5.6 Pantalla d'entrenament de VirtualDojo.....	37
Figura 5.7 Pantalla de visualització de repeticions	38
Figura 5.8 Classe Model.....	39

Figura 5.9 Classe ItemModelo.....	40
Figura 5.10 Classe Postura.....	40
Figura 5.11 Classe SkeletonInfo	41
Figura 5.12 Classe Dist.....	41
Figura 5.13 Classe CheckStance	42
Figura 6.1 Instal·lador ClickOnce	43
Figura 6.2 Configuració tipus MIME del servidor web.....	44
Figura 6.3 Configuració de Visual Studio per publicar al servidor web	45
Figura 6.4 Web d'instal·lació i suport de VirtualDojo	46
Figura 6.5 Exemple de projecte creat a Codeplex	47
Figura 7.1 Diagrama de Gantt del projecte	48
Figura 8.1 Kinect One	53

Capítol 11 *Índex de taules*

Taula 3.1 Especificacions mes rellevants de la Kinect.....	16
Taula 3.2 Característiques del SDK de Microsoft per a Kinect.....	22
Taula 7.1 Costos de Hardware	49
Taula 7.2 Costos de Software	49
Taula 7.3 Costos lloguer oficina	50
Taula 7.4 Costos de personal.....	51

Annex: Manual de VirtualDojo

1.Introducción

VirtualDojo es una aplicación para Windows que permite crear modelos de series de posturas para poder entrenarlos posteriormente y así perfeccionar técnicas de cualquier arte marcial. VirtualDojo utiliza el dispositivo Kinect de Microsoft para poder capturar el movimiento y las posturas del usuario. El programa se puede controlar mediante comandos de voz para una utilización más satisfactoria en cualquier sala de entrenamiento.

2.Requisitos del sistema

Los requisitos para el correcto funcionamiento de VirtualDojo son:

- Uno de los siguientes sistemas operativos:
 - Windows 7
 - Windows Embedded Standard 7
 - Windows 8
 - Windows Embedded Standard 8

- Requisitos de hardware
 - Procesador de 32 bits (x86) o 64 bits (x64)
 - Procesador de doble núcleo a 2,66 GHz o más
 - Bus USB 2.0 dedicado
 - 2 GB de RAM
 - Un sensor Kinect

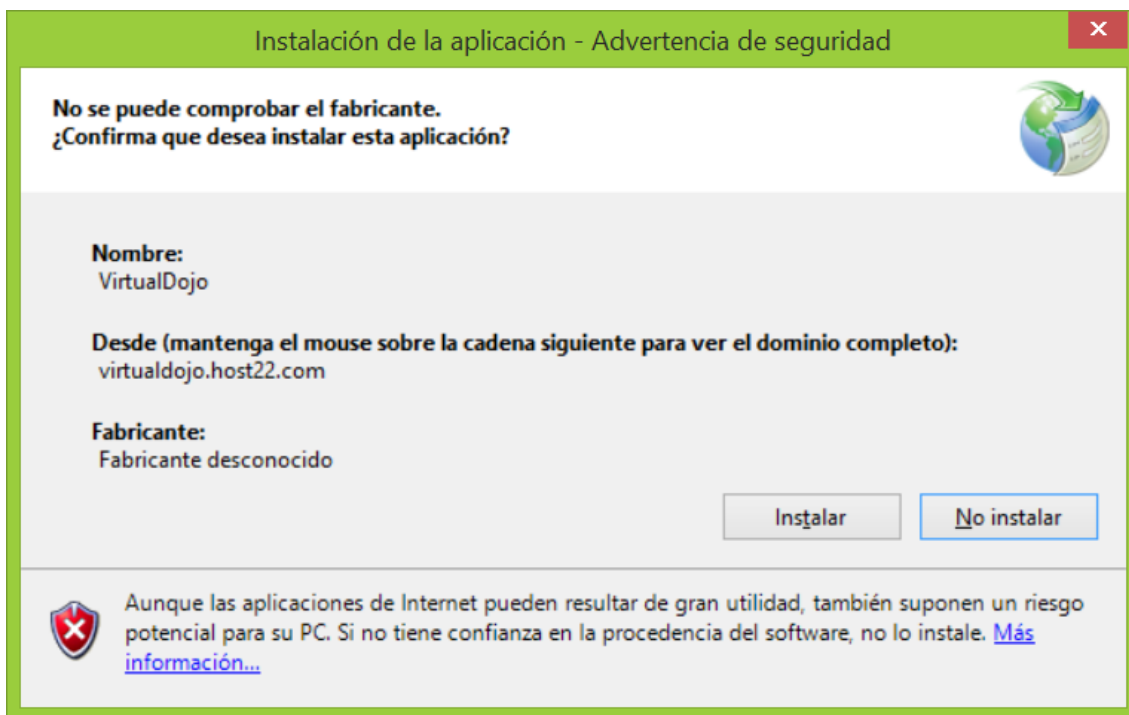
- Requisitos de software
 - [Microsoft .NET Framework 4.5 \(x86 y x64\)](#)
 - [Kinect for Windows SDK 1.8](#)
 - [Kinect for Windows Speech Recognition Language Pack \(es-ES\)](#)

Los requisitos de software pueden ser descargados desde la web de VirtualDojo:

<http://virtualdojo.host22.com/>

3.Instalación de VirtualDojo y Kinect

Al ejecutar el instalador de VirtualDojo el programa se instalará automáticamente y creará la carpeta VirtualDojo en nuestro perfil de usuario donde podremos encontrar un modelo de demostración y una repetición sobre dicho modelo.



Instalador de VirtualDojo

El programa trabaja por defecto sobre esta ruta y buscará siempre los modelos en esta carpeta.

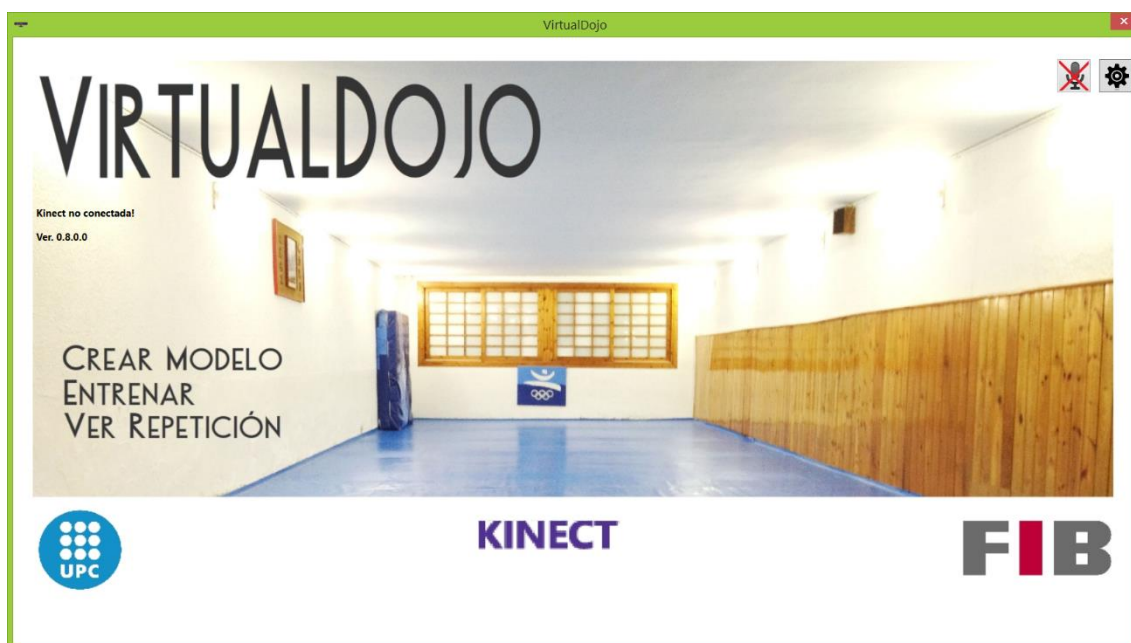
Una vez instalado el programa, si se desea, se pueden borrar el modelo y la repetición de demostración.

Si tras la instalación el programa no se inicia correctamente compruebe que en el equipo estén correctamente instalados los requerimientos de software detallados en el punto 2 de este manual.

Para el correcto funcionamiento de la aplicación la Kinect debe ver todo el cuerpo del jugador. A la hora de colocar la Kinect tenga en cuenta las siguientes consideraciones:

- Coloque el sensor cerca del borde de una superficie plana y estable.
- Coloque el sensor a una distancia comprendida entre 0,6 m y 1,8 m del suelo.
- Evite colocar el sensor en un sitio en el que le dé la luz directa del sol.
- No incline manualmente el sensor; se ajusta automáticamente o por instrucciones de voz.

4. Pantalla principal de VirtualDojo




Pantalla principal de VirtualDojo


En la pantalla principal de VirtualDojo podremos seleccionar con el ratón o con la voz los siguientes apartados:

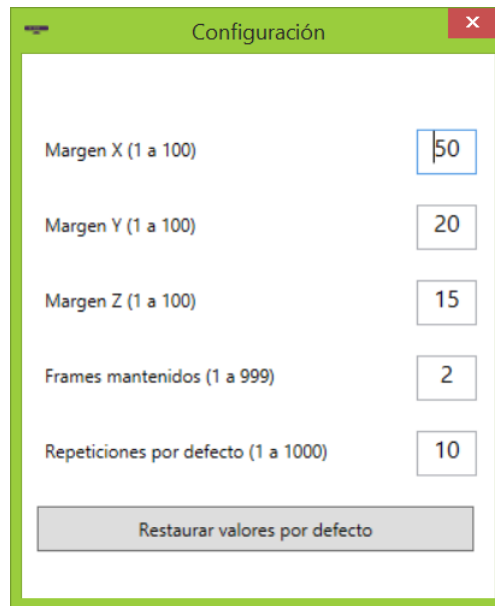
-**Crear modelo** (o con la orden "**Modelo**") nos permitirá crear un grupo de posturas para poder entrenarlas posteriormente. Estas posturas pueden formar parte de una kata o de una repetición de movimientos. Este apartado requiere de una Kinect para poder abrirlo.

-**Entrenar** (o con la orden "**Entrenar**") nos permitirá entrenar uno de los modelos creados previamente. Este apartado requiere de una Kinect para poder abrirlo.

-**Ver repetición** (o con la orden "**Repetición**") nos permitirá ver repeticiones de los entrenamientos realizados y compararlos con el modelo cargado. Este apartado no requiere de una Kinect para poder abrirlo.

Si las condiciones de la sala de entrenamiento lo requieren se podrá desactivar el reconocimiento de voz de VirtualDojo pulsando el icono 

Desde la pantalla principal también se podrán configurar distintos valores de VirtualDojo pulsando sobre el icono  que nos mostrará la siguiente ventana:



Configuración	Valor
Margen X (1 a 100)	50
Margen Y (1 a 100)	20
Margen Z (1 a 100)	15
Frames mantenidos (1 a 999)	2
Repeticiones por defecto (1 a 1000)	10

Restaurar valores por defecto

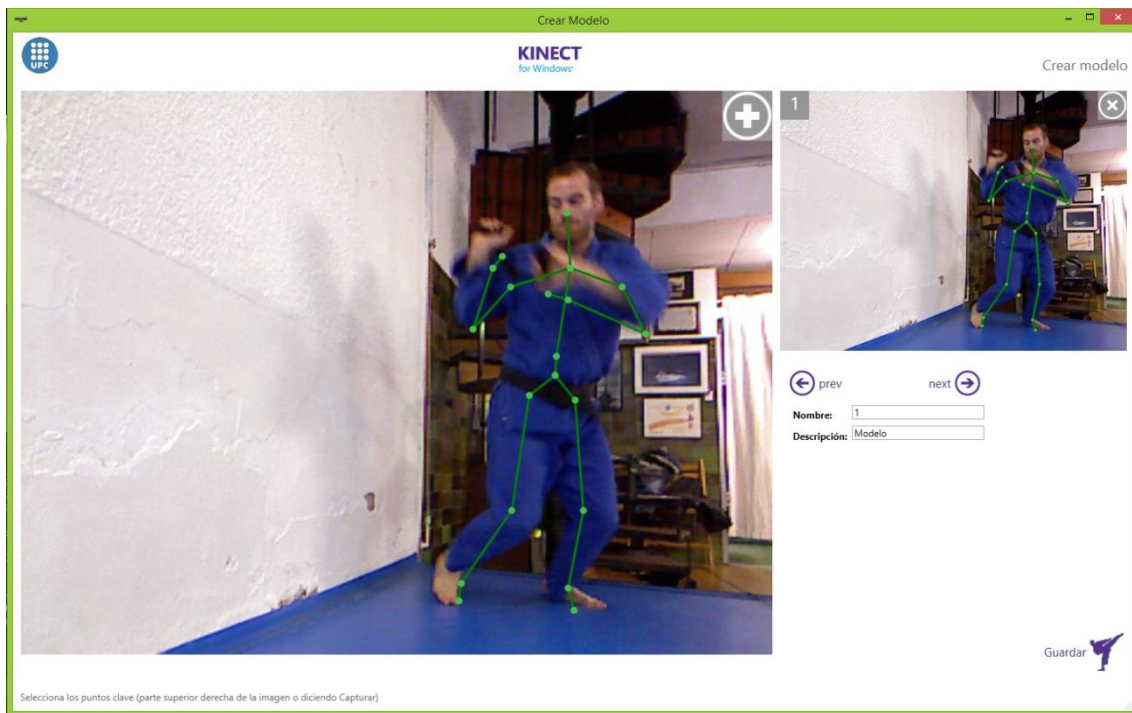
Valores configurables de VirtualDojo

Se podrán ajustar los márgenes utilizados por VirtualDojo para determinar si una postura se ha realizado correctamente. Dichos márgenes podrán contener valores entre 1 y 100. Cuanto más elevado sea el número menor será la precisión con que la que se deberán realizar las posturas. El margen por defecto para cada uno de los ejes X, Y y Z son 50, 20 y 15 respectivamente.

También es posible ajustar el número de frames que se debe mantener una postura para darla como correcta. Por defecto serán 2 frames.

Y por último también se puede configurar el número de repeticiones por defecto que mostrará VirtualDojo en el momento de seleccionar un modelo para entrenarlo o ver una repetición. Por defecto serán 10 repeticiones.


5. Crear Modelo




Ventana de creación de un modelo de posturas

En este apartado podemos crear un modelo para entrenarlo posteriormente. Disponemos de las siguientes órdenes:

- Mediante los comandos de voz "**subir**" y "**bajar**" podremos ajustar el ángulo de inclinación de Kinect.

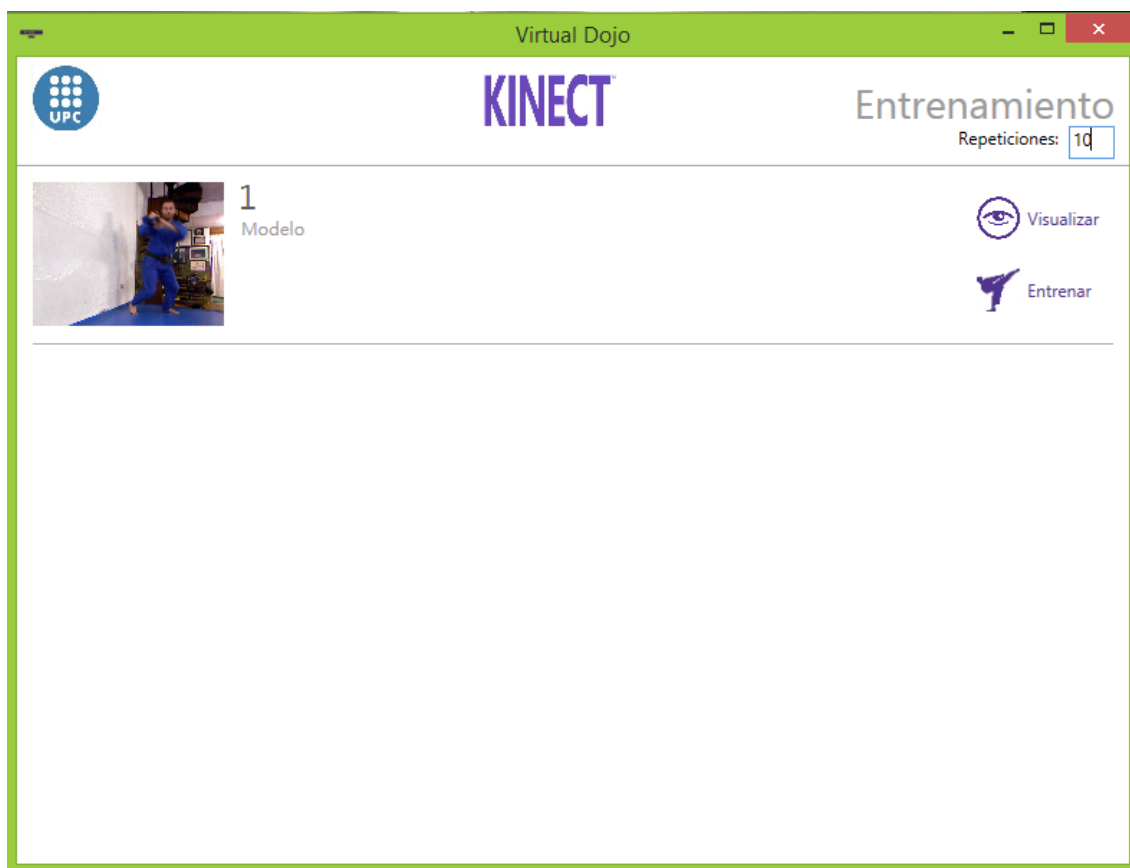
- Clicando en el icono  o mediante la orden "**capturar**" añadiremos posturas al modelo que veremos en la ventana superior derecha.

- Podremos descartar cualquier postura mediante el icono  o con la orden "**borrar**".

- Una vez terminado el modelo podemos cambiar el Nombre y la descripción del mismo o dejar los que aparecen por defecto pulsando el botón "**guardar**" o mediante la misma orden de voz.

6. Entrenar

6.1 Selección del modelo



Selección de modelos disponibles

En el apartado Entrenar primero deberemos seleccionar el modelo que queremos entrenar de la lista de modelos disponibles. Los modelos se guardan y se cargan automáticamente de la carpeta VirtualDojo de nuestro perfil de usuario.

Podemos cargar automáticamente el modelo 1 mediante el comando de voz "**entrenar**" o cualquier otro con el ratón.

En la parte superior derecha podemos seleccionar el número de repeticiones que queremos realizar del modelo seleccionado. Por defecto son 10.

6.2 Entrenamiento



Ventana de entrenamiento de VirtualDojo

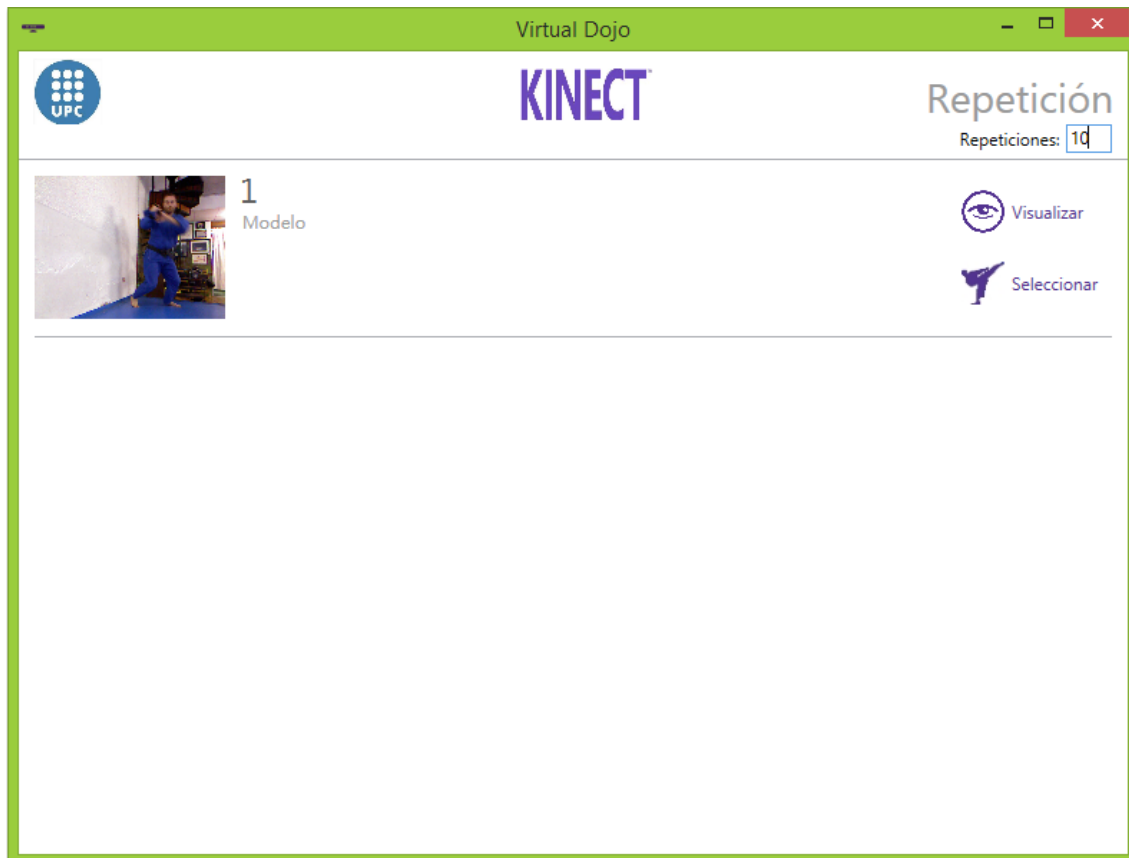
Antes de empezar el entrenamiento podemos ajustar el ángulo de la Kinect mediante los comandos de voz "**sube**", "**baja**" que aumentarán y disminuirán respectivamente el ángulo de inclinación de la Kinect.

Una vez tengamos la Kinect calibrada, si damos la orden "**Empezar**" se grabará automáticamente nuestro entrenamiento la carpeta VirtualDojo de nuestro perfil con el nombre del modelo y la extensión replay.

Una vez terminado el entrenamiento se nos preguntará si queremos renombrar la repetición. Si dejamos el nombre por defecto la siguiente vez que realicemos un entrenamiento del mismo modelo la repetición anterior se perderá.

7.Ver Repetición

7.1 Seleccionar modelo

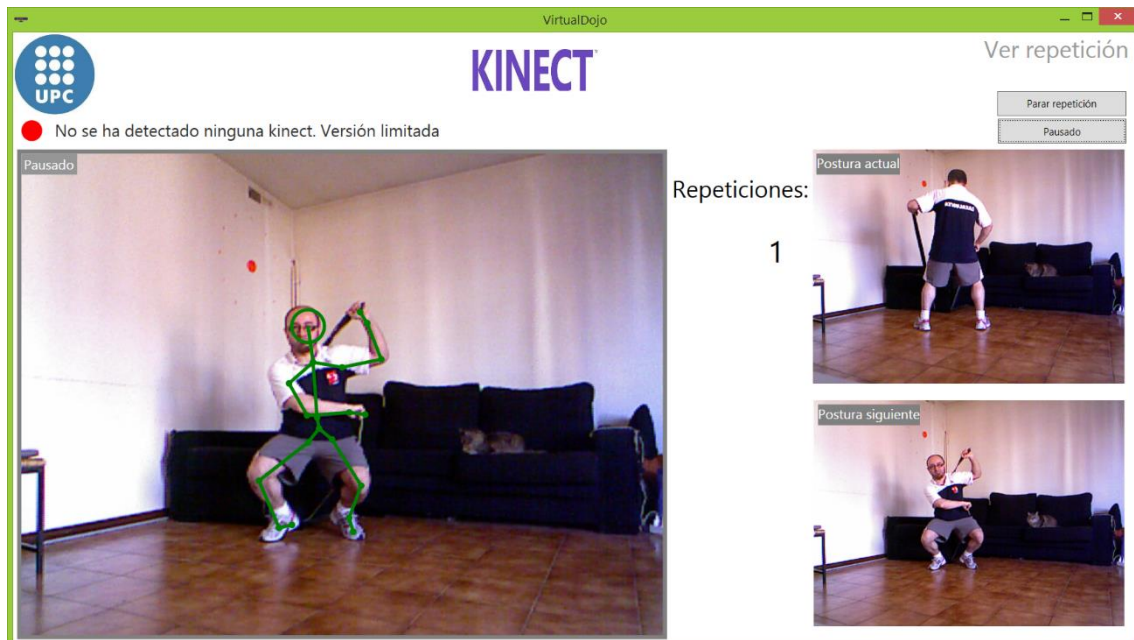


Selección de modelos disponibles para compararlos con la repetición

Del mismo modo que al realizar el entrenamiento, al entrar en el apartado de ver repeticiones debemos seleccionar el modelo con el que queremos comparar la repetición.

7.2 Reproductor de repeticiones

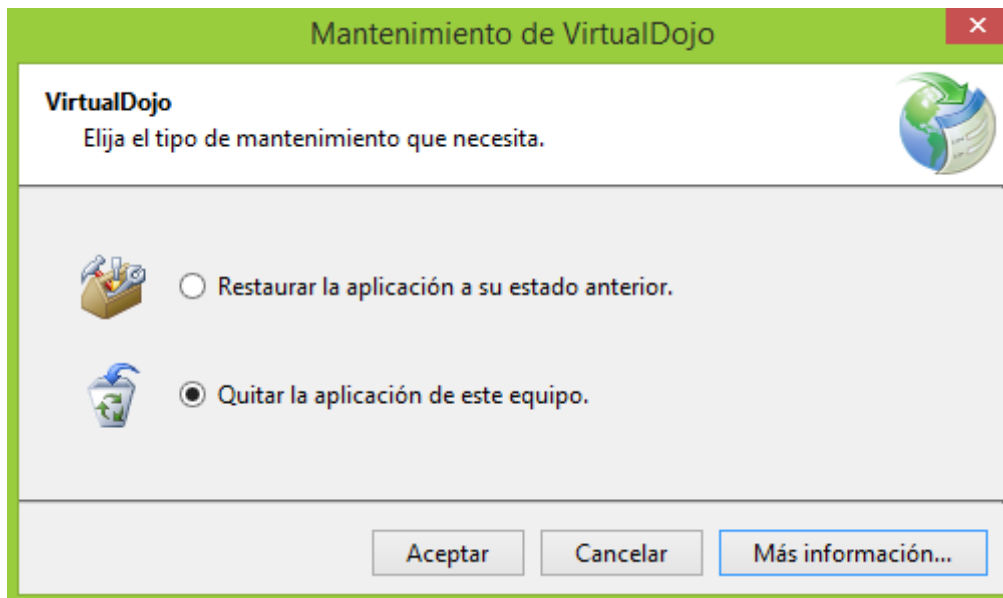
Una vez seleccionado el modelo y el número de repeticiones nos aparecerá la pantalla principal donde podremos seleccionar la repetición a cargar y podremos verla, detenerla para cargar otra o pausarla. Las repeticiones de VirtualDojo tendrán la extensión replay.



Reproductor de repeticiones de VirtualDojo

8.Desinstalación de VirtualDojo

Para desinstalar VirtualDojo debe acceder al apartado "Programas y características" del "Panel de Control" de su equipo, seleccionar la entrada de VirtualDojo de la lista de aplicaciones instaladas en su equipo y clicar en "Desinstalar o cambiar".



Ventana de desinstalación de VirtualDojo